Process Algebra with Conditionals in the Presence of Epsilon

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. In a previous paper, we presented several extensions of ACP with conditional expressions, including one with a retrospection operator on conditions to allow for looking back on conditions under which preceding actions have been performed. In this paper, we add a constant for a process that is only capable of terminating successfully to those extensions of ACP, which can be very useful in applications. It happens that in all cases the addition of this constant is unproblematic.

Keywords: empty process, retrospective conditions, condition evaluation, state operators, signal emission, splitting bisimulation, process algebra.

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

1 Introduction

In [9], we presented several extensions of ACP [7, 6] with conditional expressions. The main extensions of ACP presented in [9] are $\mathrm{ACP}^{\mathrm{c}}$, an extension of ACP with conditional expressions of the form $\zeta :\rightarrow p$ in which the conditions are taken from a free Boolean algebra over a set of generators, $\mathrm{ACP}^{\mathrm{cs}}$, an extension $\mathrm{ACP}^{\mathrm{c}}$ with a signal emission operator on processes, and $\mathrm{ACP}^{\mathrm{cr}}$, an extension of $\mathrm{ACP}^{\mathrm{c}}$ with a retrospection operator on conditions. Signal emission is usable for a special kind of condition evaluation. Retrospection allows for looking back on conditions under which preceding actions have been performed. We also extended $\mathrm{ACP}^{\mathrm{c}}$ and $\mathrm{ACP}^{\mathrm{cr}}$ with operators devised for condition evaluation and we outlined an application of $\mathrm{ACP}^{\mathrm{cr}}$ in which it allows for using conditions which express that a certain number of steps ago a certain action must have been performed.

In this paper, a constant for a process that is only capable of terminating successfully is added to the different extensions of ACP presented in [9]. This constant is often referred to as the empty process constant. In the past, the addition of the empty process constant to ACP has been treated in several ways.

The treatment in [16] yields a non-associative parallel composition operator. The first treatment that yields an associative parallel composition operator [19] is from 1986, but was not published until 1997. The addition of the empty process constant to different extensions of ACP in this paper is based on [5].

It is clear from early work [16, 19] that the addition of the empty process constant to ACP was rather problematic. Its addition to the different extensions of ACP with conditional expressions presented in [9] turns out to present no additional complications. For that reason, we look upon this paper in its current form primarily as supplementary material to [9].

The structure of this paper is as follows. First of all, we introduce $\mathrm{ACP}_\epsilon^\mathrm{c}$, the extension of ACP^c with the empty process constant (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 $\mathrm{ACP}_\epsilon^\mathrm{c}$, the main models of $\mathrm{ACP}_\epsilon^\mathrm{c}$ (Section 4). Following this, we have a closer look at splitting bisimilarity based on structural operational semantics (Section 5). Next, we extend $\mathrm{ACP}_\epsilon^\mathrm{c}$ with guarded recursion (Section 6). Thereupon, we extend $\mathrm{ACP}_\epsilon^\mathrm{c}$ with condition evaluation operators (Section 7), with state operators (Section 8) and with a signal emission operator (Section 9); and analyse how those operators are related. We also adapt the full splitting bisimulation models of $\mathrm{ACP}_\epsilon^\mathrm{c}$ to the full signal-observing splitting bisimulation models of $\mathrm{ACP}_\epsilon^\mathrm{cs}$, the extension of $\mathrm{ACP}_\epsilon^\mathrm{c}$ with signal emission (Section 10). After that, we extend $\mathrm{ACP}_\epsilon^\mathrm{c}$ with a retrospection operator (Section 11) and adapt the full splitting bisimulation models of $\mathrm{ACP}_\epsilon^\mathrm{c}$ to the full retrospective splitting bisimulation models of $\mathrm{ACP}_\epsilon^\mathrm{cr}$, the extension of $\mathrm{ACP}_\epsilon^\mathrm{c}$ with retrospection (Section 12). Thereupon, we extend $\mathrm{ACP}_\epsilon^\mathrm{cr}$ with condition evaluation operators as well (Section 13). We also outline an interesting application of $\mathrm{ACP}_\epsilon^\mathrm{cr}$ (Section 14). Finally, we make some concluding remarks (Section 15).

Some familiarity with Boolean algebras is desirable. The definitions of all notions concerning Boolean algebras that are used can be found in [17].

We thank Jan van Eijck. He communicated an application of ACP^c to us which involves a register update mechanism that cannot be dealt with in full generality without the empty process constant. This forms the greater part of our motivation to work out the addition of the empty process constant to ACP^c.

2 ACP_ϵ with Conditions

In this section, we present $\mathrm{ACP}_\epsilon^\mathrm{c}$, an extension of ACP_ϵ [5, 6] with conditional expressions of the form $\zeta{:}\rightarrow p$. $\mathrm{ACP}_\epsilon^\mathrm{c}$ can be regarded as an extension of ACP^c [9] with the empty process constant too. In $\mathrm{ACP}_\epsilon^\mathrm{c}$, as in ACP_ϵ, it is assumed that a fixed but arbitrary finite set of *actions* A, with $\delta, \epsilon \notin \mathsf{A}$, and a fixed but arbitrary commutative and associative *communication* function $|{:}\mathsf{A}_\delta \times \mathsf{A}_\delta \to \mathsf{A}_\delta$, such that $\delta \mid a = \delta$ for all $a \in \mathsf{A}_\delta$, have been given. The function $|$ is regarded to give the result of synchronously performing any two actions for which this is possible, and to be δ otherwise. Moreover, it is assumed that a fixed but arbitrary set of *atomic conditions* C_at has been given.

Let κ be an infinite cardinal. Then $\mathcal{C}_\kappa$ is the free κ-complete Boolean algebra over C_{at}.[1] As usual, we identify Boolean algebras with their domain. Thus, we also write $\mathcal{C}_\kappa$ for the domain of $\mathcal{C}_\kappa$. If κ is regular,[2] then $\mathcal{C}_\kappa$ is isomorphic to the Boolean algebra of equivalence classes with respect to logical equivalence of the set of all propositions with elements of C_{at} as propositional variables and with conjunctions and disjunctions of less than κ propositions (see e.g. [17]). In $\mathrm{ACP}_\epsilon^{\mathrm{c}}$, conditions are taken from $\mathcal{C}_{\aleph_0}$. If C_{at} is a finite set, then $\mathcal{C}_\kappa = \mathcal{C}_{\aleph_0}$ for all cardinals $\kappa > \aleph_0$. We are also interested in $\mathcal{C}_\kappa$ for cardinals $\kappa > \aleph_0$ because it permits us to consider infinitely branching processes in the case where C_{at} is an infinite set. Henceforth, we write $\mathcal{C}$ for $\mathcal{C}_{\aleph_0}$.

The algebraic theory $\mathrm{ACP}_\epsilon^{\mathrm{c}}$ has two sorts:

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

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

- the *deadlock* constant $\delta : \mathbf{P}$;
- the *empty process* constant $\epsilon : \mathbf{P}$;
- for each $a \in \mathsf{A}$, the *action* constant $a : \mathbf{P}$;
- the binary *alternative composition* operator $+ : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *sequential composition* operator $\cdot : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *guarded command* operator $:\!\to : \mathbf{C} \times \mathbf{P} \to \mathbf{P}$;
- the binary *parallel composition* operator $\parallel : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *left merge* operator $\between : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- the binary *communication merge* operator $\mid : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- for each $H \subseteq \mathsf{A}$, the unary *encapsulation* operator $\partial_H : \mathbf{P} \to \mathbf{P}$.

The algebraic theory $\mathrm{ACP}_\epsilon^{\mathrm{c}}$ 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 $\eta \in \mathsf{C}_{\mathsf{at}}$, the *atomic condition* constant $\eta : \mathbf{C}$;
- the unary *complement* operator $- : \mathbf{C} \to \mathbf{C}$;
- the binary *join* operator $\sqcup : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$;
- the binary *meet* operator $\sqcap : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$.

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 $\mathbf{P}$.

[1] For a definition of free κ-complete Boolean algebras, see e.g. [17].

[2] For a definition of regular cardinals, see e.g. [18, 13]. They include $\aleph_0$, $\aleph_1$, $\aleph_2$, ...

The constants and operators of $\mathrm{ACP}_\epsilon^\mathrm{c}$ to build terms of sort $\mathbf{P}$ are the constants and operators of ACP_ϵ and additionally the guarded command operator. Let p and q be closed terms of sort $\mathbf{P}$ and ζ and ξ be closed terms of sort $\mathbf{C}$, $a \in \mathsf{A}$, $H \subseteq \mathsf{A}$, and $\eta \in \mathsf{C}_{\mathsf{at}}$. Then, 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;
- ϵ terminates successfully, unconditionally;
- a first performs action a and then terminates successfully, both unconditionally;
- $p + q$ 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;
- $\zeta :\rightarrow p$ behaves as p under condition ζ;
- $p \parallel q$ behaves as the process that proceeds with p and q in parallel;
- $p \lfloor\!\lfloor q$ behaves the same as $p \parallel q$, except that it starts with performing an action of p;
- $p \mid q$ behaves the same as $p \parallel q$, except that it starts with performing an action of p and an action of q synchronously;
- $\partial_H(p)$ behaves the same as p, except that actions from H are blocked.

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

- η is an atomic condition;
- $\bot$ 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.

Some earlier extensions of ACP include conditional expressions of the form $p \lhd \zeta \rhd q$; see e.g. [2]. Just as in [9], 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 $\zeta :\rightarrow p + -\zeta :\rightarrow q$.

The axioms of $\mathrm{ACP}_\epsilon^\mathrm{c}$ are given in Table 1. CM3, CM7, C1–C3 and D1–D2 are actually axiom schemas in which a, b and c stand for arbitrary constants of $\mathrm{ACP}_\epsilon^\mathrm{c}$ that differ from ϵ (i.e. $a, b, c \in \mathsf{A}_\delta$). In D0–D4, H stands for an arbitrary subset of A. So, D0, D3 and D4 are axiom schemas as well. Axioms A1–A9, CM1T, TM2, CM3, CM4, TM5, TM6, CM7–CM9, C1–C3 and D0–D4 are the axioms of ACP_ϵ. Axioms BA1–BA8 are the axioms of Boolean Algebras (BA). So $\mathrm{ACP}_\epsilon^\mathrm{c}$ imports the (equational) axioms of both ACP_ϵ and BA. The axioms of BA have been taken from [15]. Several alternatives for this axiomatization can be found in the literature. Axioms GC1–GC11 have been taken from [2], but the axiom $x \cdot z \lhd \phi \rhd y \cdot z = (x \lhd \phi \rhd y) \cdot z$ (CO5) is replaced by the simpler axiom $\phi :\rightarrow x \cdot y = (\phi :\rightarrow x) \cdot y$ (GC5) and similarly for axioms GC8–GC11.

The terms of sort $\mathbf{C}$ are interpreted in $\mathcal{C}$ as usual.

Table 1. Axioms of $\mathrm{ACP}_\epsilon^{\mathrm{c}}$ $(a, b, c \in \mathsf{A}_\delta)$

$x + y = y + x$	A1	$\partial_H(\epsilon) = \epsilon$	D0
$(x + y) + z = x + (y + z)$	A2	$\partial_H(a) = a$ if $a \notin H$	D1
$x + x = x$	A3	$\partial_H(a) = \delta$ if $a \in H$	D2
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4	$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D3
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5	$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$	D4
$x + \delta = x$	A6		
$\delta \cdot x = \delta$	A7	$\top :\!\rightarrow x = x$	GC1
$x \cdot \epsilon = x$	A8	$\bot :\!\rightarrow x = \delta$	GC2
$\epsilon \cdot x = x$	A9	$\phi :\!\rightarrow \delta = \delta$	GC3
		$\phi :\!\rightarrow (x + y) = \phi :\!\rightarrow x + \phi :\!\rightarrow y$	GC4
$x \parallel y = x \between y + y \between x + x \mid y +$		$\phi :\!\rightarrow x \cdot y = (\phi :\!\rightarrow x) \cdot y$	GC5
$\partial_{\mathsf{A}}(x) \cdot \partial_{\mathsf{A}}(y)$	CM1T	$\phi :\!\rightarrow (\psi :\!\rightarrow x) = (\phi \sqcap \psi) :\!\rightarrow x$	GC6
$\epsilon \between x = \delta$	TM2	$(\phi \sqcup \psi) :\!\rightarrow x = \phi :\!\rightarrow x + \psi :\!\rightarrow x$	GC7
$a \cdot x \between y = a \cdot (x \parallel y)$	CM3	$(\phi :\!\rightarrow x) \between y = \phi :\!\rightarrow (x \between y)$	GC8
$(x + y) \between z = x \between z + y \between z$	CM4	$(\phi :\!\rightarrow x) \mid y = \phi :\!\rightarrow (x \mid y)$	GC9
$\epsilon \mid x = \delta$	TM5	$x \mid (\phi :\!\rightarrow y) = \phi :\!\rightarrow (x \mid y)$	GC10
$x \mid \epsilon = \delta$	TM6	$\partial_H(\phi :\!\rightarrow x) = \phi :\!\rightarrow \partial_H(x)$	GC11
$a \cdot x \mid b \cdot y = (a \mid b) \cdot (x \parallel y)$	CM7		
$(x + y) \mid z = x \mid z + y \mid z$	CM8	$\phi \sqcup \bot = \phi$	BA1
$x \mid (y + z) = x \mid y + x \mid z$	CM9	$\phi \sqcup -\phi = \top$	BA2
		$\phi \sqcup \psi = \psi \sqcup \phi$	BA3
$a \mid b = b \mid a$	C1	$\phi \sqcup (\psi \sqcap \chi) = (\phi \sqcup \psi) \sqcap (\phi \sqcup \chi)$	BA4
$(a \mid b) \mid c = a \mid (b \mid c)$	C2	$\phi \sqcap \top = \phi$	BA5
$\delta \mid a = \delta$	C3	$\phi \sqcap -\phi = \bot$	BA6
		$\phi \sqcap \psi = \psi \sqcap \phi$	BA7
		$\phi \sqcap (\psi \sqcup \chi) = (\phi \sqcap \psi) \sqcup (\phi \sqcap \chi)$	BA8

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

- for each $\alpha \in \mathcal{C} \setminus \{\bot\}$, a unary relation ${}^{[\alpha]}\!\downarrow$;
- for each $\ell \in (\mathcal{C} \setminus \{\bot\}) \times \mathsf{A}$, a binary relation $\xrightarrow{\ell}$.

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

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

Table 2. Transition rules for $\mathrm{ACP}_\epsilon^\mathrm{c}$

$$\frac{}{\epsilon \overset{[\top]}{\downarrow}} \qquad \frac{}{a \xrightarrow{[\top]\,a} \epsilon}$$

$$\frac{x \overset{[\phi]}{\downarrow}}{x + y \overset{[\phi]}{\downarrow}} \qquad \frac{y \overset{[\phi]}{\downarrow}}{x + y \overset{[\phi]}{\downarrow}} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x + y \xrightarrow{[\phi]\,a} x'} \qquad \frac{y \xrightarrow{[\phi]\,a} y'}{x + y \xrightarrow{[\phi]\,a} y'}$$

$$\frac{x \overset{[\phi]}{\downarrow},\ y \overset{[\psi]}{\downarrow}}{x \cdot y \overset{[\phi\sqcap\psi]}{\downarrow}} \qquad \frac{x \overset{[\phi]}{\downarrow},\ y \xrightarrow{[\psi]\,a} y'}{x \cdot y \xrightarrow{[\phi\sqcap\psi]\,a} y'}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \cdot y \xrightarrow{[\phi]\,a} x' \cdot y}$$

$$\frac{x \overset{[\phi]}{\downarrow}}{\psi :\rightarrow x \overset{[\phi\sqcap\psi]}{\downarrow}}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\psi :\rightarrow x \xrightarrow{[\phi\sqcap\psi]\,a} x'}\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \overset{[\phi]}{\downarrow},\ y \overset{[\psi]}{\downarrow}}{x \parallel y \overset{[\phi\sqcap\psi]}{\downarrow}}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \parallel y \xrightarrow{[\phi]\,a} x' \parallel y} \qquad \frac{y \xrightarrow{[\phi]\,a} y'}{x \parallel y \xrightarrow{[\phi]\,a} x \parallel y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} y'}{x \parallel y \xrightarrow{[\phi\sqcap\psi]\,c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} x'}{x \between y \xrightarrow{[\phi]\,a} x' \parallel y} \qquad \frac{x \xrightarrow{[\phi]\,a} x',\ y \xrightarrow{[\psi]\,b} y'}{x \mid y \xrightarrow{[\phi\sqcap\psi]\,c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \overset{[\phi]}{\downarrow}}{\partial_H(x) \overset{[\phi]}{\downarrow}} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\partial_H(x) \xrightarrow{[\phi]\,a} \partial_H(x')}\ a \notin H$$

The structural operational semantics of $\mathrm{ACP}_\epsilon^\mathrm{c}$ is described by the transition rules given in Table 2.

3 Transition Systems and Splitting Bisimilarity for $\mathrm{ACP}_\epsilon^\mathrm{c}$

In this section, we adapt the definitions of conditional transition systems and splitting bisimilarity of conditional transition systems from [9] to the presence of a process that is only capable of terminating successfully. In Section 4, we will make use of conditional transition systems and splitting bisimilarity of conditional transition systems as defined in this section to construct models of $\mathrm{ACP}_\epsilon^\mathrm{c}$.

The transitions of conditional transition systems have labels that consist of a condition different from $\bot$ and an action. Labels of this kind are sometimes called *guarded actions*. Henceforth, we write $\mathcal{C}_\kappa^-$ for $\mathcal{C}_\kappa \setminus \{\bot\}$.

Let κ be an infinite cardinal. Then a *κ-conditional transition system* T consists of the following:

- a set S of *states*;
- a set $\xrightarrow{\ell} \subseteq S \times S$, for each $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$;
- a set $\overset{[\alpha]}{\downarrow} \subseteq S$, for each $\alpha \in \mathcal{C}_\kappa^-$;
- an *initial state* $s^0 \in S$.

If $(s, s') \in \xrightarrow{\ell}$ for some $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$, 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 $s \, {}^{[\alpha]}\!\downarrow$ instead of $s \in {}^{[\alpha]}\!\downarrow$. Furthermore, we write $\rightarrow$ for the family of sets $(\xrightarrow{\ell})_{\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}}$ and $\downarrow$ for the family of sets $({}^{[\alpha]}\!\downarrow)_{\alpha \in \mathcal{C}_\kappa^-}$.

The relations ${}^{[\alpha]}\!\downarrow$ and $\xrightarrow{\ell}$ can be explained as follows:

- $s \, {}^{[\alpha]}\!\downarrow$: in state s, it is possible to terminate successfully under condition α;
- $s \xrightarrow{[\alpha]\, a} s'$: 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 sequence of transitions. Unreachable states, and the transitions between them, are not relevant to the behaviour represented by the transition system.

Let $T = (S, \rightarrow, \downarrow, s^0)$ be a κ-conditional transition system (for an infinite cardinal κ). 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* κ-conditional transition system if $S = \mathrm{RS}(T)$.

Henceforth, we will only consider connected conditional transition systems. However, this often calls for extraction of the connected part of a conditional transition system resulting from composition of connected conditional transition systems.

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

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

where

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

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

$$\xrightarrow{\ell}{}' = \xrightarrow{\ell} \cap\, (S' \times S') \,,$$
$${}^{[\alpha]}\!\downarrow' = {}^{[\alpha]}\!\downarrow \cap\, S' \,.$$

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 $S_1 \times S_2 \subseteq \mathcal{S}_\kappa$.[3]

Let κ be an infinite cardinal. Then $\mathbb{CTS}^\epsilon_\kappa$ is the set of all connected κ-conditional transition systems $T = (S, \rightarrow, \downarrow, s^0)$ 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 $\{(\ell, s') \in (\mathcal{C}^-_\kappa \times \mathsf{A}) \times S \mid (s, s') \in \stackrel{\ell}{\rightarrow}\} \cup \{\alpha \in \mathcal{C}^-_\kappa \mid s \in {}^{[\alpha]}\downarrow\}$ is less than κ.

The condition $S \subset \mathcal{S}_\kappa$ guarantees that $\mathbb{CTS}^\epsilon_\kappa$ 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*.

The identity of the states of a conditional transition system is not relevant to the behaviour represented by it. Conditional transition system that differ only with respect to the identity of the states are isomorphic.

Let $T_1 = (S_1, \rightarrow_1, \downarrow_1, s^0_1)$ and $T_2 = (S_2, \rightarrow_2, \downarrow_2, s^0_2)$ be κ-conditional transition systems (for an infinite cardinal κ). 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:

– $b(s^0_1) = s^0_2$;
– $s_1 \stackrel{\ell}{\rightarrow}_1 s'_1$ iff $b(s_1) \stackrel{\ell}{\rightarrow}_2 b(s'_1)$;
– $s\ {}^{[\alpha]}\downarrow_1$ iff $b(s)\ {}^{[\alpha]}\downarrow_2$.

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

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

Bisimilarity has to be adapted to the setting with guarded actions. In the definition given below, we use two well-known notions from the field of Boolean algebras: a partial order relation $\sqsubseteq$ on $\mathcal{C}_\kappa$ and a unary operation $\bigsqcup$ on the set of all subsets of $\mathcal{C}_\kappa$ of cardinality less than κ (for each infinite cardinal κ). The relation $\sqsubseteq$ and the operation $\bigsqcup$ are defined by

$$\alpha \sqsubseteq \beta \text{ iff } \alpha \sqcup \beta = \beta \qquad \text{and} \qquad \textstyle\bigsqcup C \text{ is the supremum of } C \text{ in } (\mathcal{C}_\kappa, \sqsubseteq)\ ,$$

respectively. The operation $\bigsqcup$ is defined for all subsets of $\mathcal{C}_\kappa$ of cardinality less than κ because $\mathcal{C}_\kappa$ is κ-complete.

Let $T_1 = (S_1, \rightarrow_1, \downarrow_1, s^0_1) \in \mathbb{CTS}^\epsilon_\kappa$ and $T_2 = (S_2, \rightarrow_2, \downarrow_2, s^0_2) \in \mathbb{CTS}^\epsilon_\kappa$ (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 $B(s^0_1, s^0_2)$ and for all s_1, s_2 such that $B(s_1, s_2)$:

– if $s_1 \stackrel{[\alpha]\, a}{\longrightarrow}_1 s'_1$, then there is a set $CS'_2 \subseteq \mathcal{C}^-_\kappa \times S_2$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS'_2)$ and for all $(\alpha', s'_2) \in CS'_2$, $s_2 \stackrel{[\alpha']\, a}{\longrightarrow}_2 s'_2$ and $B(s'_1, s'_2)$;

[3] 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 $\mu_2(b) = (b, \{\emptyset\})$.

– if $s_2 \xrightarrow{[\alpha]\,a}_2 s'_2$, then there is a set $CS'_1 \subseteq \mathcal{C}^-_\kappa \times S_1$ of cardinality less than κ 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 $s_1 \,{}^{[\alpha]}\!\downarrow_1$, then there is a set $C' \subseteq \mathcal{C}^-_\kappa$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \,{}^{[\alpha']}\!\downarrow_2$;
– if $s_2 \,{}^{[\alpha]}\!\downarrow_2$, then there is a set $C' \subseteq \mathcal{C}^-_\kappa$ of cardinality less than κ such that $\alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \,{}^{[\alpha']}\!\downarrow_1$.

Two conditional transition systems $T_1, T_2 \in \mathbb{CTS}^\epsilon_\kappa$ are *splitting bisimilar*, written $T_1 \leftrightarroweq T_2$, 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* $T_1 \leftrightarroweq T_2$.

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.

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

In Section 4, we will use $\mathbb{CTS}^\epsilon_\kappa$ as the domain of a structure that is a model of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$. As the domain of a structure, $\mathbb{CTS}^\epsilon_\kappa/\!\leftrightarroweq$ must be a set. That is the case because $\mathbb{CTS}^\epsilon_\kappa$ 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 $T = (S, \rightarrow, \downarrow, s^0)$ be a connected κ-conditional transition system of which the branching degree is less than κ. Then there exists a connected κ-conditional transition system $T' = (S', \rightarrow', \downarrow', s^{0\prime})$ of which the branching degree is less than κ such that $T \leftrightarroweq T'$ 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. It is also easy to see that isomorphic conditional transition systems are splitting bisimilar.

4 Full Splitting Bisimulation Models of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$

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

The models of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ are structures that consist of the following:

– a non-empty set $\mathcal{D}$, called the *domain* of the model;
– for each constant of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$, an element of $\mathcal{D}$;

– for each n-ary operator of $\mathrm{ACP}_\epsilon^\mathrm{c}$, an n-ary operation on $\mathcal{D}$.

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

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

– $\widehat{\delta} = (\{s^0\}, \emptyset, \emptyset, s^0)$,
where

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

– $\widehat{\epsilon} = (\{s^0\}, \emptyset, \downarrow, s^0)$,
where

$$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa) \ , \\ {}^{[\top]}\!\downarrow &= \{s^0\} \ , \end{aligned}$$

and for every $\alpha \in \mathcal{C}_\kappa^- \setminus \{\top\}$:

$${}^{[\alpha]}\!\downarrow = \emptyset \ .$$

– $\widehat{a} = (\{s^0, s^\surd\}, \rightarrow, \downarrow, s^0)$,
where

$$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa) \ , \\ s^\surd &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus \{s^0\}) \ , \\ \xrightarrow{[\top]\,a} &= \{(s^0, s^\surd)\} \ , \\ {}^{[\top]}\!\downarrow &= \{s^\surd\} \ , \end{aligned}$$

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

$$\begin{aligned} \xrightarrow{[\alpha']\,a'} &= \emptyset \ , \\ {}^{[\alpha'']}\!\downarrow &= \emptyset \ . \end{aligned}$$

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

$$T_1 \mathbin{\widehat{+}} T_2 = \Gamma(S, \rightarrow, \downarrow, s^0) \ ,$$

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 $(\alpha, a) \in \mathcal{C}_\kappa^- \times \mathsf{A}$ and $\alpha' \in \mathcal{C}_\kappa^-$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)} &= \{(s^0, \mu_1(s)) \mid s_1^0 \xrightarrow{[\alpha]\,a}_1 s\} \\
&\quad \cup \{(s^0, \mu_2(s)) \mid s_2^0 \xrightarrow{[\alpha]\,a}_2 s\} \\
&\quad \cup \{(\mu_1(s), \mu_1(s')) \mid s \xrightarrow{[\alpha]\,a}_1 s'\} \\
&\quad \cup \{(\mu_2(s), \mu_2(s')) \mid s \xrightarrow{[\alpha]\,a}_2 s'\} , \\
{}^{[\alpha']}\!\downarrow &= \{s^0 \mid s_1^0\ {}^{[\alpha']}\!\downarrow_1\} \\
&\quad \cup \{s^0 \mid s_2^0\ {}^{[\alpha']}\!\downarrow_2\} \\
&\quad \cup \{\mu_1(s) \mid s\ {}^{[\alpha']}\!\downarrow_1\} \\
&\quad \cup \{\mu_2(s) \mid s\ {}^{[\alpha']}\!\downarrow_2\} .
\end{aligned}$$

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

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

where

$$S = S_1 \uplus S_2 ,$$

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

$$\begin{aligned}
\xrightarrow{(\alpha,a)} &= \{(\mu_1(s), \mu_1(s')) \mid s \xrightarrow{[\alpha]\,a}_1 s' \ \wedge\ \neg\ \exists \beta \bullet s'\ {}^{[\beta]}\!\downarrow_1\} \\
&\quad \cup \{(\mu_1(s), \mu_2(s_2^0)) \mid \exists s', \beta \bullet s \xrightarrow{[\alpha]\,a}_1 s' \ \wedge\ s'\ {}^{[\beta]}\!\downarrow_1\} \\
&\quad \cup \{(\mu_2(s_2^0), \mu_2(s')) \mid \\
&\qquad\qquad \exists s, \beta, \beta' \bullet s\ {}^{[\beta]}\!\downarrow_1 \wedge\ s_2^0 \xrightarrow{[\beta']\,a}_2 s' \ \wedge\ \alpha = \beta \sqcap \beta'\} \\
&\quad \cup \{(\mu_2(s), \mu_2(s')) \mid s \xrightarrow{[\alpha]\,a}_2 s' \ \wedge\ s \neq s_2^0\} , \\
{}^{[\alpha']}\!\downarrow &= \{\mu_2(s_2^0) \mid \exists s, \beta, \beta' \bullet s\ {}^{[\beta]}\!\downarrow_1 \wedge\ s_2^0\ {}^{[\beta']}\!\downarrow_2 \wedge\ \alpha' = \beta \sqcap \beta'\} \\
&\quad \cup \{\mu_2(s) \mid s\ {}^{[\alpha']}\!\downarrow_2 \wedge\ s \neq s_2^0\} .
\end{aligned}$$

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

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

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

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

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

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

where

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

$$S\ = S_1 \times S_2\ ,$$

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

$$\begin{aligned}
\xrightarrow{(\alpha,a)} &= \{((s_1,s_2),(s_1',s_2)) \mid s_1 \xrightarrow{[\alpha]\,a}_1 s_1' \ \wedge\ s_2 \in S_2\} \\
&\cup \{((s_1,s_2),(s_1,s_2')) \mid s_1 \in S_1 \ \wedge\ s_2 \xrightarrow{[\alpha]\,a}_2 s_2'\} \\
&\cup \{((s_1,s_2),(s_1',s_2')) \mid \\
&\qquad \bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-,\, a',b' \in \mathsf{A}} (s_1 \xrightarrow{[\alpha']\,a'}_1 s_1' \ \wedge\ s_2 \xrightarrow{[\beta']\,b'}_2 s_2' \ \wedge \\
&\qquad\qquad \alpha' \sqcap \beta' = \alpha \ \wedge\ a' \mid b' = a)\}\ ,
\end{aligned}$$

$$^{[\alpha'']}{\downarrow}\ = \{(s_1,s_2) \mid \bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-} (s_1\ {}^{[\alpha']}{\downarrow}_1 \wedge\ s_2\ {}^{[\beta']}{\downarrow}_2 \wedge\ \alpha' \sqcap \beta' = \alpha'')\}\ .$$

– Let $T_i = (S_i, \rightarrow_i, \downarrow_i, s_i^0) \in \mathbb{CTS}_\kappa^\epsilon$ for $i = 1, 2$. Suppose that $T_1 \mathbin{\widehat{\|}} T_2 = (S, \rightarrow, \downarrow, s^0)$. Then

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

where

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

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

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

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

– Let $T_i = (S_i, \rightarrow_i, \downarrow_i, s_i^0) \in \mathbb{CTS}_\kappa^\epsilon$ for $i = 1, 2$. Suppose that $T_1 \mathbin{\widehat{\|}} T_2 = (S, \rightarrow, \downarrow, s^0)$. Then

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

where

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

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

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

$$\begin{aligned}
\xrightarrow{(\alpha,a)}{}' = \{(s^{0\prime},(s_1,s_2)) \mid \\
\bigvee_{\alpha',\beta' \in \mathcal{C}_\kappa^-,\, a',b' \in \mathsf{A}} (s_1^0 \xrightarrow{[\alpha']\,a'}_1 s_1 \ \wedge\ s_2^0 \xrightarrow{[\beta']\,b'}_2 s_2 \ \wedge \\
\alpha' \sqcap \beta' = \alpha \ \wedge\ a' \mid b' = a)\}\ .
\end{aligned}$$

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

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

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

$$\xrightarrow{(\alpha,a)}{}' = \xrightarrow{(\alpha,a)} \ ,$$

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

$$\xrightarrow{(\alpha,a)}{}' = \emptyset \ .$$

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

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

We can show that splitting bisimilarity is a congruence with respect to the operations on $\mathbb{CTS}^\epsilon_\kappa$ associated with the operators of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$.

Proposition 4.1 (Congruence). *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}^\epsilon_\kappa$ and $\alpha \in \mathcal{C}_\kappa$, $T_1 \leftrightarroweq T_1'$ and $T_2 \leftrightarroweq T_2'$ imply $T_1 \mathbin{\widehat{+}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{+}} T_2'$, $T_1 \mathbin{\widehat{\cdot}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{\cdot}} T_2'$, $\alpha :\widehat{\rightarrow}\, T_1 \leftrightarroweq \alpha :\widehat{\rightarrow}\, T_1'$, $T_1 \mathbin{\widehat{\parallel}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{\parallel}} T_2'$, $T_1 \mathbin{\widehat{\between}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{\between}} T_2'$, $T_1 \mathbin{\widehat{\between}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{\between}} T_2'$, $T_1 \mathbin{\widehat{\between}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{\between}} T_2'$ and $\widehat{\partial_H}(T_1) \leftrightarroweq \widehat{\partial_H}(T_1')$.*

Proof. For all operations except $\widehat{\parallel}$, witnessing splitting bisimulations are constructed in the same way as in the congruence proofs for the corresponding operations on $\mathbb{CTS}_\kappa$ given in [9]. For $\widehat{\parallel}$, the construction of a witnessing splitting bisimulation is easier than in [9].[4] Let R_1 and R_2 be splitting bisimulations witnessing $T_1 \leftrightarroweq T_1'$ and $T_2 \leftrightarroweq T_2'$, respectively. Then we construct relations $R_{\widehat{\parallel}}$ as follows:

– $R_{\widehat{\parallel}} = \{((s_1, s_2), (s_1', s_2')) \mid (s_1, s_1') \in R_1, (s_2, s_2') \in R_2\}$.

Given the definition of parallel composition, it is easy to see that $R_{\widehat{\parallel}}$ is a splitting bisimulation witnessing $T_1 \mathbin{\widehat{\parallel}} T_2 \leftrightarroweq T_1' \mathbin{\widehat{\parallel}} T_2'$. □

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

[4] Because the relation constructed in [9] is by mistake the same as the one constructed in this paper, we should actually say "in the revision of [9] that can be found at `www.win.tue.nl/~keesm/sbrc.pdf`".

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

where those ingredients are defined as follows:

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

The operations on $\mathbb{CTS}^{\epsilon}_{\kappa}/\leftrightarrows$ are well-defined because $\leftrightarrows$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}^{\epsilon}_{\kappa}$.

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

Theorem 4.1 (Soundness of $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$). *For each infinite cardinal κ, we have $\mathfrak{P}^{\epsilon c}_{\kappa} \models \mathrm{ACP}^{\mathrm{c}}_{\epsilon}$.*

Proof. Because $\leftrightarrows$ is a congruence, it is sufficient to show that all additional axioms are sound. The soundness of all additional axioms follows easily from the definition of $\mathfrak{P}^{\epsilon c}_{\kappa}$. □

For all axioms that are in common with $\mathrm{ACP}^{\mathrm{c}}$, the proof of soundness with respect to $\mathfrak{P}^{\epsilon c}_{\kappa}$ follows the same line as the proof of soundness with respect to $\mathfrak{P}^{\mathrm{c}}_{\kappa}$.

The full splitting bisimulation models are related by isomorphic embeddings.

Theorem 4.2 (Isomorphic Embedding). *Let κ and κ' be infinite cardinals such that $\kappa < \kappa'$. Then $\mathfrak{P}^{\epsilon c}_{\kappa}$ is isomorphically embedded in $\mathfrak{P}^{\epsilon c}_{\kappa'}$.*

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 [9]. □

5 SOS-Based Splitting Bisimilarity for $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$

It is customary to associate transition systems with closed terms of 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 splitting bisimilar.

The structural operational semantics of $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$ presented in Section 2 determines a conditional transition system for each process that can be denoted by

a closed term of sort $\mathbf{P}$. These transition systems are special in the sense that their states are closed terms of sort $\mathbf{P}$.

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

- S is the set of all closed terms of sort $\mathbf{P}$;
- the sets $\xrightarrow{(\alpha,a)} \subseteq S \times S$ and ${}^{[\alpha]}\downarrow \subseteq S$ 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 2 hold;
- $s^0 \in S$ is the closed term p.

Let p and q be closed terms of sort $\mathbf{P}$. Then we say that p and q are *splitting bisimilar*, written $p \leftrightarrows q$, if $\mathrm{CTS}(p) \leftrightarrows \mathrm{CTS}(q)$.

Clearly, the structural operational semantics does not give rise to infinitely branching conditional transition systems. For each closed term p of sort $\mathbf{P}$, there exists a $T \in \mathbb{CTS}^\epsilon_{\aleph_0}$ 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 Guarded Recursion

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

A *recursive specification* over $\mathrm{ACP}_\epsilon^\mathrm{c}$ is a set of 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 $\mathbf{P}$ 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 $\mathrm{ACP}_\epsilon^\mathrm{c}$) $\{P_X \mid X \in \mathrm{V}(E)\}$ 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 $\mathbf{P}$ 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 $\mathrm{ACP}_\epsilon^\mathrm{c}$ 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 $\mathrm{ACP}_\epsilon^\mathrm{c}$ and the equations of the recursive specification. We are only interested in models of $\mathrm{ACP}_\epsilon^\mathrm{c}$ 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 $\mathbf{P}$ and E be a guarded recursive specification over $\mathrm{ACP}_\epsilon^\mathrm{c}$. 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 the equations given in Table 3. Both

Table 3. 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

Table 4. Transition rules for recursion

$$\frac{\langle t_X|E\rangle \stackrel{[\phi]}{\downarrow}}{\langle X|E\rangle \stackrel{[\phi]}{\downarrow}} X{=}t_X \in E \qquad \frac{\langle t_X|E\rangle \xrightarrow{[\phi]\,a} x'}{\langle X|E\rangle \xrightarrow{[\phi]\,a} x'} X{=}t_X \in E$$

RDP and RSP are axiom schemas. A side condition is 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.

The structural operational semantics for the constants $\langle X|E\rangle$ is described by the transition rules given in Table 4.

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

Theorem 6.1 (Unique solutions in $\mathfrak{P}^{\epsilon\mathrm{c}}_{\kappa}$). *For each infinite cardinal κ, guarded recursive specifications over $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$ have unique solutions in $\mathfrak{P}^{\epsilon\mathrm{c}}_{\kappa}$.*

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 [9]. □

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

7 Evaluation of Conditions

Guarded commands cannot always be eliminated from closed terms of sort $\mathbf{P}$ because conditions different from both $\bot$ and $\top$ may be involved. The condition evaluation operators introduced below, can be brought into action in such cases. These operators require to fix an infinite cardinal λ. By doing so, full splitting bisimulation models with domain $\mathbb{CTS}^{\epsilon}_{\kappa}/\leftrightarrows$ for $\kappa > \lambda$ are excluded.

There are unary λ-*complete condition evaluation* operators $\mathsf{CE}_h : \mathbf{P} \to \mathbf{P}$ and $\mathsf{CE}_h : \mathbf{C} \to \mathbf{C}$ for each λ-complete endomorphisms h of $\mathcal{C}_\lambda$.[5]

These operators can be explained as follows: $\mathsf{CE}_h(p)$ behaves as p with each condition ζ occurring in p replaced according to h. If the image of $\mathcal{C}_\lambda$ under h

[5] For a definition of κ-complete endomorphisms, see e.g. [17].

Table 5. Axioms for condition evaluation ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$, $\eta' \in \mathsf{C}_{\mathsf{at}} \cup \{\bot, \top\}$)

$\mathsf{CE}_h(\epsilon) = \epsilon$	CE1T	$\mathsf{CE}_h(\bot) = \bot$	CE6
$\mathsf{CE}_h(a \cdot x) = a \cdot \mathsf{CE}_h(x)$	CE2	$\mathsf{CE}_h(\top) = \top$	CE7
$\mathsf{CE}_h(x + y) = \mathsf{CE}_h(x) + \mathsf{CE}_h(y)$	CE3	$\mathsf{CE}_h(\eta) = \eta'$ if $h(\eta) = \eta'$	CE8
$\mathsf{CE}_h(\phi :\rightarrow x) = \mathsf{CE}_h(\phi) :\rightarrow \mathsf{CE}_h(x)$	CE4	$\mathsf{CE}_h(-\phi) = -\mathsf{CE}_h(\phi)$	CE9
$\mathsf{CE}_h(\mathsf{CE}_{h'}(x)) = \mathsf{CE}_{h \circ h'}(x)$	CE5	$\mathsf{CE}_h(\phi \sqcup \psi) = \mathsf{CE}_h(\phi) \sqcup \mathsf{CE}_h(\psi)$	CE10
		$\mathsf{CE}_h(\phi \sqcap \psi) = \mathsf{CE}_h(\phi) \sqcap \mathsf{CE}_h(\psi)$	CE11

Table 6. Transition rules for condition evaluation

$$\frac{x \overset{[\phi]}{\downarrow}}{\mathsf{CE}_h(x) \overset{[h(\phi)]}{\downarrow}} \; h(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{\mathsf{CE}_h(x) \xrightarrow{[h(\phi)]\, a} \mathsf{CE}_h(x')} \; h(\phi) \neq \bot$$

is $\mathbb{B}$, i.e. the Boolean algebra with domain $\{\bot, \top\}$, then guarded commands can be eliminated from $\mathsf{CE}_h(p)$. In the case where the image of $\mathcal{C}_\lambda$ under h is not $\mathbb{B}$, CE_h can be regarded to evaluate the conditions only partially.

Henceforth, we write $\mathcal{H}_\lambda$ for the set of all λ-complete endomorphisms of $\mathcal{C}_\lambda$.

The additional axioms for CE_h, where $h \in \mathcal{H}_\lambda$, are the axioms given in Table 5.

The structural operational semantics of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ extended with condition evaluation is described by the transition rules for $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ and the transition rules given in Table 6.

If λ is a regular infinite cardinal, the elements of $\mathcal{C}_\lambda$ can be used to represent equivalence classes with respect to logical equivalence of the set of all propositions with elements of C_{at} as propositional variables and with conjunctions and disjunctions of less than λ propositions. We write $\mathcal{P}_\lambda$ for this set of propositions. If λ is a regular infinite cardinal, it is likely that there is a theory Φ about the atomic conditions in the shape of a set of propositions. Let $\Phi \subset \mathcal{P}_\lambda$, and let $h_\Phi \in \mathcal{H}_\lambda$ be such that for all $\alpha, \beta \in \mathcal{C}_\lambda$:

$$\Phi \vdash \langle\!\langle h_\Phi(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle \alpha \rangle\!\rangle \quad \text{and} \quad h_\Phi(\alpha) = h_\Phi(\beta) \text{ iff } \Phi \vdash \langle\!\langle \alpha \rangle\!\rangle \Leftrightarrow \langle\!\langle \beta \rangle\!\rangle \qquad (1)$$

where $\langle\!\langle \alpha \rangle\!\rangle$ is a representative of the equivalence class of propositions isomorphic to α. Then we have $h_\Phi(\alpha) = \top$ iff $\langle\!\langle \alpha \rangle\!\rangle$ is derivable from Φ and $h_\Phi(\alpha) = \bot$ iff $\neg \langle\!\langle \alpha \rangle\!\rangle$ is derivable from Φ. The image of $\mathcal{C}_\lambda$ under h_Φ is $\mathbb{B}$ iff Φ is a complete theory. If Φ is not a complete theory, then h_Φ is not uniquely determined by (1). However, the images of $\mathcal{C}_\lambda$ under the different endomorphisms satisfying (1) are isomorphic subalgebras of $\mathcal{C}_\lambda$. Moreover, if both h and h' satisfy (1), then $\Phi \vdash \langle\!\langle h(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle h'(\alpha) \rangle\!\rangle$ for all $\alpha \in \mathcal{C}_\lambda$.

Below, we show that condition evaluation on the basis of a complete theory can be viewed as substitution on the basis of the theory. That leads us to the use of the following convention: for $\alpha \in \mathcal{C}$, $\underline{\alpha}$ stands for an arbitrary closed term of sort $\mathbf{C}$ of which the value in $\mathcal{C}$ is α.

Proposition 7.1 (Condition evaluation on the basis of a theory). *Assume that λ is a regular infinite cardinal. Let $\Phi \subset \mathcal{P}_\lambda$ be a complete theory and let p be a closed term of sort $\mathbf{P}$. Then $\mathsf{CE}_{h_\Phi}(p) = p'$ where p' is p with, for all $\alpha \in \mathcal{C}$, in all subterms of the form $\underline{\alpha} :\rightarrow q$, $\underline{\alpha}$ replaced by $\top$ if $\Phi \vdash \langle\!\langle \alpha \rangle\!\rangle$ and $\underline{\alpha}$ replaced by $\bot$ if $\Phi \vdash \neg \langle\!\langle \alpha \rangle\!\rangle$.*

Proof. This result follows immediately from the definition of h_Φ and the distributivity of CE_{h_Φ} over all operators of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$. □

In μCRL [14], an extension of ACP which includes conditional expressions, we find a formalization of the substitution-based alternative for CE_{h_Φ}.

The substitution-based alternative works properly because condition evaluation by means of a λ-complete condition evaluation operator is not dependent on process behaviour. Hence, the result of condition evaluation is globally valid. Below, we will generalize the condition evaluation operators introduced above in such a way that condition evaluation may be dependent on process behaviour. In that case, the result of condition evaluation is in general not globally valid.

Remark 7.1. Assume that λ is a regular infinite cardinal. Let $h \in \mathcal{H}_\lambda$. Then h induces a theory $\Phi \subset \mathcal{P}_\lambda$ such that $h = h_\Phi$, viz. the theory Φ defined by

$$\Phi = \{\langle\!\langle h(\alpha) \rangle\!\rangle \Leftrightarrow \langle\!\langle \alpha \rangle\!\rangle \mid \alpha \in \mathcal{C}_\lambda\} \cup \{\langle\!\langle \alpha \rangle\!\rangle \Leftrightarrow \langle\!\langle \beta \rangle\!\rangle \mid h(\alpha) = h(\beta)\} \ .$$

Consequently, if λ is a regular infinite cardinal, condition evaluation by means of the λ-complete condition evaluation operators introduced above is always condition evaluation of which the result can be determined from a set of propositions. We will return to this observation in Section 9.

We proceed with generalizing the condition evaluation operators introduced above. It is assumed that a fixed but arbitrary function $\mathsf{eff} : \mathsf{A} \times \mathcal{H}_\lambda \to \mathcal{H}_\lambda$ has been given.

There is a unary *generalized λ-complete condition evaluation* operator $\mathsf{GCE}_h : \mathbf{P} \to \mathbf{P}$ for each $h \in \mathcal{H}_\lambda$; and there is again the unary operator $\mathsf{CE}_h : \mathbf{C} \to \mathbf{C}$ for each $h \in \mathcal{H}_\lambda$.

The λ-complete generalized condition evaluation operator GCE_h allows, given the function eff, to evaluate conditions dependent of process behaviour. The function eff gives, for each action a and λ-complete endomorphism h, the λ-complete endomorphism h' that represents the changed results of condition evaluation due to performing a. The function eff is extended to A_δ such that $\mathsf{eff}(\delta, h) = h$ for all $h \in \mathcal{H}_\lambda$.

The additional axioms for GCE_h, where $h \in \mathcal{H}_\lambda$, are the axioms given in Table 7 and axioms CE6–CE11 from Table 5.

The structural operational semantics of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ extended with generalized condition evaluation is described by the transition rules for $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ and the transition rules given in Table 8.

We can add both the λ-complete condition evaluation operators and the generalized λ-complete condition evaluation operators to $\mathrm{ACP}^{\mathrm{c}}_\epsilon$. However, Proposition 7.2 stated below makes it clear that the latter operators supersede the former operators.

Table 7. Axioms for generalized condition evaluation ($a \in \mathsf{A}_\delta$)

$\mathsf{GCE}_h(\epsilon) = \epsilon$	GCE1T
$\mathsf{GCE}_h(a \cdot x) = a \cdot \mathsf{GCE}_{\mathsf{eff}(a,h)}(x)$	GCE2
$\mathsf{GCE}_h(x + y) = \mathsf{GCE}_h(x) + \mathsf{GCE}_h(y)$	GCE3
$\mathsf{GCE}_h(\phi :\rightarrow x) = \mathsf{CE}_h(\phi) :\rightarrow \mathsf{GCE}_h(x)$	GCE4

Table 8. Transition rules for generalized condition evaluation

$$\frac{x \stackrel{[\phi]}{\downarrow}}{\mathsf{GCE}_h(x) \stackrel{[h(\phi)]}{\downarrow}} \; h(\phi) \neq \bot \qquad \frac{x \stackrel{[\phi]\,a}{\longrightarrow} x'}{\mathsf{GCE}_h(x) \stackrel{[h(\phi)]\,a}{\longrightarrow} \mathsf{GCE}_{\mathsf{eff}(a,h)}(x')} \; h(\phi) \neq \bot$$

The full splitting bisimulation models of $\mathrm{ACP}_\epsilon^\mathrm{c}$ with condition evaluation and/or generalized condition evaluation are simply the expansions of the full splitting bisimulation models $\mathfrak{P}_\kappa^{\mathrm{ec}}$ of $\mathrm{ACP}_\epsilon^\mathrm{c}$, for infinite cardinals $\kappa \leq \lambda$, obtained by associating with each operator CE_h and/or GCE_h the corresponding re-labeling operation on conditional transition systems. As mentioned before, full splitting bisimulation models with domain $\mathbb{CTS}_\kappa^\epsilon/\leftrightarroweq$ for $\kappa > \lambda$ are excluded.

The equation $\mathsf{CE}_h(\mathsf{CE}_{h'}(x)) = \mathsf{CE}_{h \circ h'}(x)$ is an axiom, but the equation $\mathsf{GCE}_h(\mathsf{GCE}_{h'}(x)) = \mathsf{GCE}_{h \circ h'}(x)$ is not an axiom. The reason is that the latter equation is only valid if eff satisfies $\mathsf{eff}(a, h \circ h') = \mathsf{eff}(a, h) \circ \mathsf{eff}(a, h')$ for all $a \in \mathsf{A}$ and $h, h' \in \mathcal{H}_\lambda$.

As their name suggests, the generalized λ-complete condition evaluation operators are generalizations of the λ-complete condition evaluation operators.

Proposition 7.2 (Generalization). *We can fix the function* eff *such that* $\mathsf{GCE}_h(x) = \mathsf{CE}_h(x)$ *for all* $h \in \mathcal{H}_\lambda$.

Proof. Clearly, if $\mathsf{eff}(a, h') = h'$ for all $a \in \mathsf{A}$ and $h' \in \mathcal{H}_\lambda$, then $\mathsf{GCE}_h(x) = \mathsf{CE}_h(x)$ for all $h \in \mathcal{H}_\lambda$. □

The λ-complete state operators that are added to $\mathrm{ACP}_\epsilon^\mathrm{c}$ in Section 8 are in their turn generalizations of the generalized λ-complete condition evaluation operators.

We come back to the λ-complete condition evaluation operators CE_h for $h \in \mathcal{H}_\lambda$. The image of $\mathcal{C}_\lambda$ under the λ-complete endomorphism h is a subalgebra of $\mathcal{C}_\lambda$ that is λ-complete too. For that reason, we could have used λ-complete homomorphisms to subalgebras that are λ-complete instead of λ-complete endomorphisms. It would go beyond the models of the theory developed so far to generalize this in such a way that λ-complete homomorphisms to λ-complete Boolean algebras different from subalgebras of $\mathcal{C}_\lambda$ are also included.

However, in the case where we consider λ-complete homomorphisms between free λ-complete Boolean algebras over different sets of generators, we can relate the models for different choices for C_{at}.

Let C and C' be different choices for $\mathsf{C_{at}}$,[6] and let $\mathfrak{P}^{\epsilon c}_{\kappa}(C)$ and $\mathfrak{P}^{\epsilon c}_{\kappa}(C')$, for $\kappa \leq \lambda$, be the full splitting bisimulation models $\mathfrak{P}^{\epsilon c}_{\kappa}$ of $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$ for the different choices for $\mathsf{C_{at}}$. Moreover, let h be a λ-complete homomorphism from the free λ-complete Boolean algebra over C to the free λ-complete Boolean algebra over C'. Then h can be extended to a homomorphism h^* from $\mathfrak{P}^{\epsilon c}_{\kappa}(C)$ to $\mathfrak{P}^{\epsilon c}_{\kappa}(C')$. This homomorphism is defined by

$$h^*([\,(S, \rightarrow, \downarrow, s^0)\,]_{\leftrightarrows}) = [\,\Gamma(S, \rightarrow', \downarrow', s^0)\,]_{\leftrightarrows}\,,$$

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

$$\begin{aligned} \xrightarrow{(\alpha,a)}{}' &= \{(s, s') \mid \exists \beta \bullet s \xrightarrow{[\beta]\, a} s' \wedge \alpha = h(\beta)\}\,, \\ {}^{[\alpha']}\!\downarrow' &= \{s \mid \exists \beta \bullet s\, {}^{[\beta]}\!\downarrow \wedge\ \alpha' = h(\beta)\}\,. \end{aligned}$$

It is easy to see that h^* is well-defined and a homomorphism indeed.

Thus, a λ-complete homomorphism between λ-complete Boolean algebras over different sets of generators can be used to translate conditions throughout a full splitting bisimulation model for one choice of $\mathsf{C_{at}}$ in such a way that a full splitting bisimulation model for a different choice of $\mathsf{C_{at}}$ is obtained.

8 State Operators

The state operators make it easy to represent the execution of a process in a state. The basic idea is that the execution of an action in a state has effect on the state, i.e. it causes a change of state. Besides, there is an action left when an action is executed in a state. The operators introduced here generalize the state operators added to ACP in [1]. The main difference with those operators is that guarded commands are taken into account. As in the case of the condition evaluation operators and the generalized condition evaluation operators, these state operators require to fix an infinite cardinal λ. By doing so, full splitting bisimulation models with domain $\mathbb{CTS}^{\epsilon}_{\kappa}/\leftrightarrows$ for $\kappa > \lambda$ are excluded.

It is assumed that a fixed but arbitrary set S of *states* has been given, together with functions $\mathsf{act} : \mathsf{A} \times S \to \mathsf{A}_\delta$, $\mathsf{eff} : \mathsf{A} \times S \to S$ and $\mathsf{eval} : \mathcal{C}_\lambda \times S \to \mathcal{C}_\lambda$, where, for each $s \in S$, the function $h_s : \mathcal{C}_\lambda \to \mathcal{C}_\lambda$ defined by $h_s(\alpha) = \mathsf{eval}(\alpha, s)$ is a λ-complete endomorphism of $\mathcal{C}_\lambda$.

There are unary λ-*complete state operators* $\lambda_s : \mathbf{P} \to \mathbf{P}$ and $\lambda_s : \mathbf{C} \to \mathbf{C}$ for each $s \in S$.[7]

The λ-complete state operator λ_s allows, given the above-mentioned functions, processes to interact with a state. Let p be a process. Then $\lambda_s(p)$ is the process p executed in state s. The function act gives, for each action a and state s, the action that results from executing a in state s. The function eff gives, for

[6] The interesting cases are those where the cardinalities of C and C' are different. Otherwise, the homomorphisms are isomorphisms.

[7] Holding on to the usual conventions leads to the double use of the symbol λ: without subscript it stands for an infinite cardinal, and with subscript it stands for a state operator.

Table 9. Axioms for state operators ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$, $\eta' \in \mathsf{C}_{\mathsf{at}} \cup \{\bot, \top\}$)

$\lambda_s(\epsilon) = \epsilon$	SO1T	$\lambda_s(\bot) = \bot$	SO5
$\lambda_s(a \cdot x) = \mathsf{act}(a,s) \cdot \lambda_{\mathsf{eff}(a,s)}(x)$	SO2	$\lambda_s(\top) = \top$	SO6
$\lambda_s(x + y) = \lambda_s(x) + \lambda_s(y)$	SO3	$\lambda_s(\eta) = \eta'$ if $\mathsf{eval}(\eta, s) = \eta'$	SO7
$\lambda_s(\phi :\rightarrow x) = \lambda_s(\phi) :\rightarrow \lambda_s(x)$	SO4	$\lambda_s(-\phi) = -\lambda_s(\phi)$	SO8
		$\lambda_s(\phi \sqcup \psi) = \lambda_s(\phi) \sqcup \lambda_s(\psi)$	SO9
		$\lambda_s(\phi \sqcap \psi) = \lambda_s(\phi) \sqcap \lambda_s(\psi)$	SO10

Table 10. Transition rules for state operators

$$\frac{x \overset{[\phi]}{\downarrow}}{\lambda_s(x) \overset{[\mathsf{eval}(\phi,s)]}{\downarrow}} \; \mathsf{eval}(\phi, s) \neq \bot$$

$$\frac{x \overset{[\phi]\, a}{\longrightarrow} x'}{\lambda_s(x) \overset{[\mathsf{eval}(\phi,s)]\, \mathsf{act}(a,s)}{\longrightarrow} \lambda_{\mathsf{eff}(a,s)}(x')} \; \mathsf{act}(a, s) \neq \delta,\ \mathsf{eval}(\phi, s) \neq \bot$$

each action a and state s, the state that results from executing a in state s. The function eval gives, for each condition α and state s, the condition that results from evaluating α in state s. The functions act and eff are extended to A_δ such that $\mathsf{act}(\delta, s) = \delta$ and $\mathsf{eff}(\delta, s) = s$ for all $s \in S$.

The additional axioms for λ_s, where $s \in S$, are the axioms given in Table 9.

The structural operational semantics of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ extended with state operators is described by the transition rules for $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ and the transition rules given in Table 10.

The full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ with state operators are simply the expansions of the full splitting bisimulation models $\mathfrak{P}^{\mathrm{ec}}_\kappa$ of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ obtained by associating with each operator λ_s the corresponding re-labeling operation on conditional transition systems.

We can add, in addition to the λ-complete state operators, the λ-complete condition evaluation operators and/or the generalized λ-complete condition evaluation operators from Section 7 to $\mathrm{ACP}^{\mathrm{c}}_\epsilon$.

We write $\mathfrak{P}^{\mathrm{ec}\,\mathrm{ext}}_\kappa$ for the expansion of $\mathfrak{P}^{\mathrm{ec}}_\kappa$ for the λ-complete condition evaluation operators, the generalized λ-complete condition evaluation operators and the λ-complete state operators.

The λ-complete state operators are generalizations of the generalized λ-complete condition evaluation operators from Section 7.

Proposition 8.1 (Generalization). *We can fix S, act, eff and eval such that, for some $f{:}\mathcal{H}_\lambda \to S$, $\lambda_{f(h)}(x) = \mathsf{GCE}_h(x)$ holds for all $h \in \mathcal{H}_\lambda$ in all full splitting bisimulation models $\mathfrak{P}^{\mathrm{ec}\,\mathrm{ext}}_\kappa$ with $\kappa \leq \lambda$.*

Proof. Clearly, if $S = \mathcal{H}_\lambda$, f is the identity function on $\mathcal{H}_\lambda$, and $\mathsf{act}(a, s) = a$, $\mathsf{eff}(a, s) = \mathsf{eff}(a, f^{-1}(s))$ and $\mathsf{eval}(\alpha, s) = f^{-1}(s)(\alpha)$ for all $a \in \mathsf{A}$, $s \in S$ and

$\alpha \in \mathcal{C}_\lambda$, then $\lambda_{f(h)}(x) = \mathsf{GCE}_h(x)$ holds for all $h \in \mathcal{H}_\lambda$ in all full splitting bisimulation models $\mathfrak{P}_\kappa^{\mathrm{ec}\,\mathrm{ext}}$ with $\kappa \leq \lambda$. $\square$

9 Signal Emission

In Section 7, we made the observation that, if λ is a regular infinite cardinal, condition evaluation by means of the λ-complete condition evaluation operators CE_h from that section is always condition evaluation of which the result can be determined from a set of propositions (see Remark 7.1). A similar observation can be made about condition evaluation by means of the generalized λ-complete condition evaluation operators GCE_h from that section. In the case of condition evaluation by means of CE_h, the set of propositions determining the result of condition evaluation does not change as a process proceeds. In the case of condition evaluation by means of GCE_h, it may happen that the set of propositions determining the result of condition evaluation changes as a process proceeds. That is, the sets of propositions relevant to a process and its subprocesses may differ. This suggest that condition evaluation can also be dealt with by explicitly associating sets of propositions with processes. The intuition is, then, that all propositions from the set of propositions associated with a process holds at the start of the process.

Clearly, if we restrict ourselves to sets of propositions of cardinality less than a regular infinite cardinal λ, we can associate elements of $\mathcal{C}_\lambda$ with processes instead. In line with [2], the element of $\mathcal{C}_\lambda$ associated with a process is called the signal emitted by the process. Because $\bot$ represents the proposition F, the proposition that cannot hold at the start of any process, we regard a process with which $\bot$ is associated as an inconsistency. However, in an algebraic setting, we cannot exclude this inconsistency. Therefore, we consider it to be a special process, which is called the inaccessible process.[8]

The idea to associate elements of $\mathcal{C}_\lambda$ with processes naturally suggests itself in the case where λ is a regular infinite cardinal. However, there are no trammels to drop the restriction that λ is regular.

All this leads us to an extension of $\mathrm{ACP}_\epsilon^{\mathrm{c}}$, called $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$, with the following additional constants and operators:

- the *inaccessible process* constant $\bot : \mathbf{P}$;
- the binary *signal emission* operator $\between : \mathbf{C} \times \mathbf{P} \to \mathbf{P}$.

The axioms of $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$ are the axioms of $\mathrm{ACP}_\epsilon^{\mathrm{c}}$ with axioms CM2, CM3 and GC8–GC10 replaced by axioms CM2ST, CM3S and GC8S–GC10S from Table 11, and the additional axioms given in Table 12. Axioms NE1–NE3 and SE1–SE11 have been taken from [3] and axioms GC9S and GC10S have been taken from [3] with subterms of the form $\mathsf{s}(x) \between \delta$ replaced by $\partial_{\mathsf{A}}(x)$. Axioms

[8] In [12, 8], this process is rather contradictory called the non-existent process. Its new name was prompted by the fact that after performing an action no process will ever proceed as this process.

Table 11. Axioms adapted to signal emission ($a \in \mathsf{A}_\delta$)

$\epsilon \between x = \partial_{\mathsf{A}}(x)$	CM2ST
$a \cdot x \between y = a \cdot (x \parallel y) + \partial_{\mathsf{A}}(y)$	CM3S
$(\phi :\rightarrow x) \between y = \phi :\rightarrow (x \between y) + \partial_{\mathsf{A}}(y)$	GC8S
$(\phi :\rightarrow x) \mid y = \phi :\rightarrow (x \mid y) + \partial_{\mathsf{A}}(y)$	GC9S
$x \mid (\phi :\rightarrow y) = \phi :\rightarrow (x \mid y) + \partial_{\mathsf{A}}(x)$	GC10S

Table 12. Additional axioms for signal emission ($a \in \mathsf{A}_\delta$)

$x + \bot = \bot$	NE1	$\phi \mathbin{\curlywedge} (\psi \mathbin{\curlywedge} x) = (\phi \sqcap \psi) \mathbin{\curlywedge} x$	SE5
$\bot \cdot x = \bot$	NE2	$\phi \mathbin{\curlywedge} (\phi :\rightarrow x) = \phi \mathbin{\curlywedge} x$	SE6
$a \cdot \bot = \delta$	NE3	$\phi :\rightarrow (\psi \mathbin{\curlywedge} x) = (-\phi \sqcup \psi) \mathbin{\curlywedge} (\phi :\rightarrow x)$	SE7
$\top \mathbin{\curlywedge} x = x$	SE1	$(\phi \mathbin{\curlywedge} x) \between y = \phi \mathbin{\curlywedge} (x \between y)$	SE8
$\bot \mathbin{\curlywedge} x = \bot$	SE2	$(\phi \mathbin{\curlywedge} x) \mid y = \phi \mathbin{\curlywedge} (x \mid y)$	SE9
$\phi \mathbin{\curlywedge} x + y = \phi \mathbin{\curlywedge} (x + y)$	SE3	$x \mid (\phi \mathbin{\curlywedge} y) = \phi \mathbin{\curlywedge} (x \mid y)$	SE10
$(\phi \mathbin{\curlywedge} x) \cdot y = \phi \mathbin{\curlywedge} x \cdot y$	SE4	$\partial_H(\phi \mathbin{\curlywedge} x) = \phi \mathbin{\curlywedge} \partial_H(x)$	SE11

CM2ST, CM3S and GC8S differ really from the corresponding axioms in [3] due to the choice of having as the signal emitted by the left merge of two processes, as in the case of the communication merge, always the meet of the signals emitted by the two processes.

In the structural operational semantics of $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$, unary relations s^α, one for each $\alpha \in \mathcal{C} \setminus \{\bot\}$, are used in addition to the relations ${}^{[\alpha]}\!\downarrow$ and $\xrightarrow{\ell}$. We write $\mathsf{s}(p) = \alpha$ instead of $p \in \mathsf{s}^\alpha$. The relation s^α can be explained as follows:

- $\mathsf{s}(p) = \alpha$: p emits the signal α.

The structural operational semantics of $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$ is described by the transition rules given in Table 13. These transition rules include all transition rules from Table 2 with additional premises to exclude transitions from or to processes that emit the signal $\bot$. There are additional transition rules describing the signals emitted by the processes. The transition rules for signal emission are new as well.

The following gives a good picture of the nature of signals and conditions.

Proposition 9.1 (Signals and conditions). *If* $\langle\langle \alpha \rangle\rangle \vdash \langle\langle \beta \rangle\rangle \Leftrightarrow \langle\langle \beta' \rangle\rangle$, *then* $\underline{\alpha} \mathbin{\curlywedge} (\underline{\beta} :\rightarrow x) = \underline{\alpha} \mathbin{\curlywedge} (\underline{\beta'} :\rightarrow x)$.

Proof. The proof is the same to the proof of the corresponding proposition in the setting of $\mathrm{ACP}^{\mathrm{cs}}$ given in [9]. □

We have the following corollaries from Proposition 9.1.

Corollary 9.1. *If* $\langle\langle \alpha \rangle\rangle \vdash \langle\langle \beta \rangle\rangle$, *then* $\underline{\alpha} \mathbin{\curlywedge} (\underline{\beta} :\rightarrow x) = \underline{\alpha} \mathbin{\curlywedge} x$. *If* $\langle\langle \alpha \rangle\rangle \vdash \neg \langle\langle \beta \rangle\rangle$, *then* $\underline{\alpha} \mathbin{\curlywedge} (\underline{\beta} :\rightarrow x) = \underline{\alpha} \mathbin{\curlywedge} \delta$.

Table 13. Transition rules for $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$

$$\frac{}{\epsilon\ {}^{[\top]}\!\downarrow} \qquad \frac{}{a \xrightarrow{[\top]\ a} \epsilon}$$

$$\frac{x\ {}^{[\phi]}\!\downarrow,\ \mathsf{s}(x+y) \neq \bot}{x+y\ {}^{[\phi]}\!\downarrow} \qquad \frac{y\ {}^{[\phi]}\!\downarrow,\ \mathsf{s}(x+y) \neq \bot}{x+y\ {}^{[\phi]}\!\downarrow}$$

$$\frac{x \xrightarrow{[\phi]\ a} x',\ \mathsf{s}(x+y) \neq \bot}{x+y \xrightarrow{[\phi]\ a} x'} \qquad \frac{y \xrightarrow{[\phi]\ a} y',\ \mathsf{s}(x+y) \neq \bot}{x+y \xrightarrow{[\phi]\ a} y'}$$

$$\frac{x\ {}^{[\phi]}\!\downarrow,\ y\ {}^{[\phi]}\!\downarrow}{x \cdot y\ {}^{[\phi\sqcap\psi]}\!\downarrow} \qquad \frac{x\ {}^{[\phi]}\!\downarrow,\ y \xrightarrow{[\psi]\ a} y'}{x \cdot y \xrightarrow{[\phi\sqcap\psi]\ a} y'}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\ a} x'}{x \cdot y \xrightarrow{[\phi]\ a} x' \cdot y}$$

$$\frac{x\ {}^{[\phi]}\!\downarrow}{\psi :\!\rightarrow x\ {}^{[\phi\sqcap\psi]}\!\downarrow}\ \phi \sqcap \psi \neq \bot \qquad \frac{x \xrightarrow{[\phi]\ a} x'}{\psi :\!\rightarrow x \xrightarrow{[\phi\sqcap\psi]\ a} x'}\ \phi \sqcap \psi \neq \bot$$

$$\frac{x\ {}^{[\phi]}\!\downarrow,\ \mathsf{s}(\psi \between x) \neq \bot}{\psi \between x\ {}^{[\phi]}\!\downarrow} \qquad \frac{x \xrightarrow{[\phi]\ a} x',\ \mathsf{s}(\psi \between x) \neq \bot}{\psi \between x \xrightarrow{[\phi]\ a} x'}$$

$$\frac{x\ {}^{[\phi]}\!\downarrow,\ y\ {}^{[\psi]}\!\downarrow,\ \mathsf{s}(x \parallel y) \neq \bot}{x \parallel y\ {}^{[\phi\sqcap\psi]}\!\downarrow}\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\ a} x',\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x' \parallel y) \neq \bot}{x \parallel y \xrightarrow{[\phi]\ a} x' \parallel y} \qquad \frac{y \xrightarrow{[\phi]\ a} y',\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x \parallel y') \neq \bot}{x \parallel y \xrightarrow{[\phi]\ a} x \parallel y'}$$

$$\frac{x \xrightarrow{[\phi]\ a} x',\ y \xrightarrow{[\psi]\ b} y',\ \mathsf{s}(x \parallel y) \neq \bot,\ \mathsf{s}(x' \parallel y') \neq \bot}{x \parallel y \xrightarrow{[\phi\sqcap\psi]\ c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\ a} x',\ \mathsf{s}(x \between y) \neq \bot,\ \mathsf{s}(x' \parallel y) \neq \bot}{x \between y \xrightarrow{[\phi]\ a} x' \parallel y}$$

$$\frac{x \xrightarrow{[\phi]\ a} x',\ y \xrightarrow{[\psi]\ b} y',\ \mathsf{s}(x \mid y) \neq \bot,\ \mathsf{s}(x' \parallel y') \neq \bot}{x \mid y \xrightarrow{[\phi\sqcap\psi]\ c} x' \parallel y'}\ a \mid b = c,\ \phi \sqcap \psi \neq \bot$$

$$\frac{x\ {}^{[\phi]}\!\downarrow}{\partial_H(x)\ {}^{[\phi]}\!\downarrow} \qquad \frac{x \xrightarrow{[\phi]\ a} x'}{\partial_H(x) \xrightarrow{[\phi]\ a} \partial_H(x')}\ a \notin H$$

$$\frac{}{\mathsf{s}(\bot) = \bot} \qquad \frac{}{\mathsf{s}(\epsilon) = \top} \qquad \frac{}{\mathsf{s}(a) = \top} \qquad \frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x+y) = \phi \sqcap \psi}$$

$$\frac{\mathsf{s}(x) = \phi}{\mathsf{s}(x \cdot y) = \phi} \qquad \frac{\mathsf{s}(x) = \phi}{\mathsf{s}(\psi :\!\rightarrow y) = -\psi \sqcup \phi} \qquad \frac{\mathsf{s}(x) = \phi}{\mathsf{s}(\psi \between y) = \psi \sqcap \phi}$$

$$\frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x \parallel y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x \between y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi,\ \mathsf{s}(y) = \psi}{\mathsf{s}(x \mid y) = \phi \sqcap \psi} \qquad \frac{\mathsf{s}(x) = \phi}{\mathsf{s}(\partial_H(x)) = \phi}$$

Corollary 9.2. *If* $\mathsf{eff}(h, a)$ *is the identity endomorphism on* $\mathcal{C}$ *for all endomorphisms* h *on* $\mathcal{C}$ *and* $a \in \mathsf{A}$*, then we have* $\mathsf{GCE}_{h_{\{\langle\langle\alpha\rangle\rangle\}}}(\underline{\beta} :\!\rightarrow x) = \underline{\beta'} :\!\rightarrow \mathsf{GCE}_{h_{\{\langle\langle\alpha\rangle\rangle\}}}(x)$ *implies* $\underline{\alpha} \between (\underline{\beta} :\!\rightarrow x) = \underline{\alpha} \between (\underline{\beta'} :\!\rightarrow x)$.

10 Full Signal-Observing Splitting Bisimulation Models of $\mathrm{ACP}_{\epsilon}^{\mathrm{cs}}$

In this section, we introduce conditional transition systems with signals, signal-observing splitting bisimilarity of conditional transition systems with signals, and the full signal-observing splitting bisimulation models of $\mathrm{ACP}_{\epsilon}^{\mathrm{cs}}$.

Conditional transition systems with signals generalize conditional transition systems.

Let κ be an infinite cardinal. Then a *κ-conditional transition system with signals* T is a tuple $(S, \rightarrow, \downarrow, \mathsf{s}, s^0)$ where

- $(S, \rightarrow, \downarrow, s^0)$ is a κ-conditional transition system;
- s is a function from S to $\mathcal{C}_\kappa$;

and for all $\ell \in \mathcal{C}_\kappa^- \times \mathsf{A}$ and $\alpha \in \mathcal{C}_\kappa^-$:

- $\{(s, s') \in \xrightarrow{\ell} \mid \mathsf{s}(s) = \bot \vee \mathsf{s}(s') = \bot\} = \emptyset$;
- $\{s \in {}^{[\alpha]}\!\downarrow \mid \mathsf{s}(s) = \bot\} = \emptyset$.

We say that $\mathsf{s}(s)$ is the signal emitted by the state s.

For conditional transition systems with signals, reachability and connectedness are defined exactly as for conditional transition systems.

Let $(S, \rightarrow, \downarrow, \mathsf{s}, s^0)$ be a κ-conditional transition system with signals (for an infinite cardinal κ) that is not necessarily connected. Then the *connected part* of T, written $\Gamma(T)$, is simply defined as follows:

$$\Gamma(T) = (S', \rightarrow', \downarrow', \mathsf{s}', s^0) \ ,$$

where

$$(S', \rightarrow', \downarrow', s^0) = \Gamma(S, \rightarrow, \downarrow, s^0) \ ,$$
$$\mathsf{s}' \text{ is the restriction of } \mathsf{s} \text{ to } S' \ .$$

Let κ be an infinite cardinal. Then $\mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ is the set of all κ-conditional transition systems with signals $(S, \rightarrow, \downarrow, \mathsf{s}, s^0)$ for which $(S, \rightarrow, \downarrow, s^0) \in \mathbb{CTS}_\kappa^{\epsilon}$.

Isomorphism between conditional transition systems with signals is defined as between conditional transition systems, but with the additional condition that $\mathsf{s}_1(s) = \mathsf{s}_2(b(s))$. Splitting bisimilarity has to be adapted to the setting with signals.

Let $T_1 = (S_1, \rightarrow_1, \downarrow_1, \mathsf{s}_1, s_1^0) \in \mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$, $T_2 = (S_2, \rightarrow_2, \downarrow_2, \mathsf{s}_2, s_2^0) \in \mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ (for an infinite cardinal κ). Then a *signal-observing 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 $B(s_1, s_2)$:

- $\mathsf{s}_1(s_1) = \mathsf{s}_2(s_2)$;
- if $s_1 \xrightarrow{[\alpha]\, a}_1 s_1'$, then there is a set $CS_2' \subseteq \mathcal{C}_\kappa^- \times S_2$ of cardinality less than κ such that $\mathsf{s}_1(s_1) \sqcap \alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_2')$ and for all $(\alpha', s_2') \in CS_2'$, $s_2 \xrightarrow{[\alpha']\, a}_2 s_2'$ and $B(s_1', s_2')$;

– if $s_2 \xrightarrow{[\alpha]\,a}_2 s_2'$, then there is a set $CS_1' \subseteq \mathcal{C}_\kappa^- \times S_1$ of cardinality less than κ such that $\mathsf{s}_2(s_2) \sqcap \alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_1')$ and for all $(\alpha', s_1') \in CS_1'$, $s_1 \xrightarrow{[\alpha']\,a}_1 s_1'$ and $B(s_1', s_2')$;
– if $s_1 \,{}^{[\alpha]}\!\downarrow_1$, then there is a set $C' \subseteq \mathcal{C}_\kappa^-$ of cardinality less than κ such that $\mathsf{s}_1(s_1) \sqcap \alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \,{}^{[\alpha']}\!\downarrow_2$;
– if $s_2 \,{}^{[\alpha]}\!\downarrow_2$, then there is a set $C' \subseteq \mathcal{C}_\kappa^-$ of cardinality less than κ such that $\mathsf{s}_2(s_2) \sqcap \alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \,{}^{[\alpha']}\!\downarrow_1$.

Two conditional transition systems with signals $T_1, T_2 \in \mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ are *signal-observing splitting bisimilar*, written $T_1 \leftrightarroweq^{\mathrm{s}} T_2$, if there exists a signal-observing splitting bisimulation B between T_1 and T_2. Let B be a signal-observing splitting bisimulation between T_1 and T_2. Then we say that B is a splitting signal-observing bisimulation *witnessing* $T_1 \leftrightarroweq^{\mathrm{s}} T_2$.

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

The elements of $\mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ and operations on $\mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ to be associated with the constants and operators of $\mathrm{ACP}_\epsilon^{\mathrm{c}}$ are as the elements of $\mathbb{CTS}_\kappa^{\epsilon}$ and operations on $\mathbb{CTS}_\kappa^{\epsilon}$ associated with them, but with all relations ${}^{[\alpha]}\!\downarrow$ and $\xrightarrow{\ell}$ restricted to states that emit a signal different from $\bot$ and with the additional function s as suggested by the structural operational semantics of $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$.

We associate with the additional constant $\bot$ an element $\widehat{\bot}^{\mathrm{s}}$ of $\mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ and with the additional operator $\mathbin{\triangleleft}$ an operation $\mathbin{\widehat{\triangleleft}^{\mathrm{s}}}$ on $\mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ as follows.

– $\widehat{\bot}^{\mathrm{s}} = (\{s^0\}, \emptyset, \emptyset, \mathsf{s}, s^0)$,
where

$$\mathsf{s}(s^0) = \bot \ .$$

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

$$\alpha \mathbin{\widehat{\triangleleft}^{\mathrm{s}}} T = \Gamma(S, \rightarrow', \downarrow', \mathsf{s}', s^0) \ ,$$

where

$$\begin{aligned} \mathsf{s}'(s) &= \mathsf{s}(s) && \text{for } s \in S \setminus \{s^0\} \ , \\ \mathsf{s}'(s^0) &= \alpha \sqcap \mathsf{s}(s^0) \ , \end{aligned}$$

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

$$\begin{aligned} \xrightarrow{(\alpha,a)}{}' &= \{(s, s') \mid s \xrightarrow{[\alpha]\,a} s' \wedge \mathsf{s}'(s) \neq \bot \wedge \mathsf{s}'(s') \neq \bot\} \ , \\ {}^{[\alpha']}\!\downarrow' &= \{s \mid s \,{}^{[\alpha']}\!\downarrow \wedge\ \mathsf{s}'(s) \neq \bot\} \ . \end{aligned}$$

We can easily show that signal-observing splitting bisimilarity is a congruence with respect to the operations on $\mathbb{CTS}_\kappa^{\epsilon\mathrm{s}}$ associated with the operators of $\mathrm{ACP}_\epsilon^{\mathrm{cs}}$.

Proposition 10.1 (Congruence). *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}^{\epsilon}_{\kappa}$ and $\alpha \in \mathcal{C}_\kappa$, $T_1 \leftrightarrows^{\mathrm{s}} T_1'$ and $T_2 \leftrightarrows^{\mathrm{s}} T_2'$ imply $T_1 \mathbin{\widehat{+}}^{\mathrm{s}} T_2 \leftrightarrows^{\mathrm{s}} T_1' \mathbin{\widehat{+}}^{\mathrm{s}} T_2'$, $T_1 \mathbin{\widehat{\cdot}}^{\mathrm{s}} T_2 \leftrightarrows^{\mathrm{s}} T_1' \mathbin{\widehat{\cdot}}^{\mathrm{s}} T_2'$, $\alpha :\widehat{\rightarrow}^{\mathrm{s}} T_1 \leftrightarrows^{\mathrm{s}} \alpha :\widehat{\rightarrow}^{\mathrm{s}} T_1'$, $\alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1 \leftrightarrows^{\mathrm{s}} \alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1'$, $T_1 \mathbin{\widehat{\parallel}}^{\mathrm{s}} T_2 \leftrightarrows^{\mathrm{s}} T_1' \mathbin{\widehat{\parallel}}^{\mathrm{s}} T_2'$, $T_1 \mathbin{\widehat{\between}}^{\mathrm{s}} T_2 \leftrightarrows^{\mathrm{s}} T_1' \mathbin{\widehat{\between}}^{\mathrm{s}} T_2'$, $T_1 \mathbin{\widehat{\upharpoonright}}^{\mathrm{s}} T_2 \leftrightarrows^{\mathrm{s}} T_1' \mathbin{\widehat{\upharpoonright}}^{\mathrm{s}} T_2'$ and $\widehat{\partial_H}^{\mathrm{s}}(T_1) \leftrightarrows^{\mathrm{s}} \widehat{\partial_H}^{\mathrm{s}}(T_1')$.*

Proof. For $\mathbin{\widehat{+}}^{\mathrm{s}}$, $\mathbin{\widehat{\cdot}}^{\mathrm{s}}$, $:\widehat{\rightarrow}^{\mathrm{s}}$, $\mathbin{\widehat{\parallel}}^{\mathrm{s}}$, $\mathbin{\widehat{\between}}^{\mathrm{s}}$, $\mathbin{\widehat{\upharpoonright}}^{\mathrm{s}}$ and $\widehat{\partial_H}^{\mathrm{s}}$, witnessing signal-observing splitting bisimulations are constructed in the same way as witnessing splitting bisimulations are constructed in the proof of Proposition 4.1. What remains is to construct a witnessing signal-observing splitting bisimulation for $\mathbin{\widehat{\triangleleft}}^{\mathrm{s}}$. Let R be a signal-observing splitting bisimulation witnessing $T_1 \leftrightarrows^{\mathrm{s}} T_1'$. Then we construct a relation $R_{\mathbin{\widehat{\triangleleft}}^{\mathrm{s}}}$ as follows:

- $R_{\mathbin{\widehat{\triangleleft}}^{\mathrm{s}}} = R \cap (S \times S')$, where S and S' are the sets of states of $\alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1$ and $\alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1'$, respectively.

Given the definition of signal emission, it is easy to see that $R_{\mathbin{\widehat{\triangleleft}}^{\mathrm{s}}}$ is a signal-observing splitting bisimulation witnessing $\alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1 \leftrightarrows^{\mathrm{s}} \alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1'$. □

The ingredients of the *full signal-observing splitting bisimulation models* $\mathfrak{P}^{\epsilon\mathrm{cs}}_{\kappa}$ of $\mathrm{ACP}^{\mathrm{cs}}_{\epsilon}$, one for each infinite cardinal κ, are defined as follows:

$$\begin{aligned}
&\mathcal{P} = \mathbb{CTS}^{\epsilon\mathrm{s}}_{\kappa}/\leftrightarrows^{\mathrm{s}}\,,\\
&\widetilde{\bot}^{\mathrm{s}} = [\,\widehat{\bot}^{\mathrm{s}}\,]_{\leftrightarrows^{\mathrm{s}}}\,, && \alpha :\widetilde{\rightarrow}^{\mathrm{s}} [\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} = [\,\alpha :\widehat{\rightarrow}^{\mathrm{s}} T_1\,]_{\leftrightarrows^{\mathrm{s}}}\,,\\
&\widetilde{\delta}^{\mathrm{s}} = [\,\widehat{\delta}^{\mathrm{s}}\,]_{\leftrightarrows^{\mathrm{s}}}\,, && \alpha \mathbin{\widetilde{\triangleleft}}^{\mathrm{s}} [\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} = [\,\alpha \mathbin{\widehat{\triangleleft}}^{\mathrm{s}} T_1\,]_{\leftrightarrows^{\mathrm{s}}}\,,\\
&\widetilde{\epsilon}^{\mathrm{s}} = [\,\widehat{\epsilon}^{\mathrm{s}}\,]_{\leftrightarrows^{\mathrm{s}}}\,, && [\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} \mathbin{\widetilde{\parallel}}^{\mathrm{s}} [\,T_2\,]_{\leftrightarrows^{\mathrm{s}}} = [\,T_1 \mathbin{\widehat{\parallel}}^{\mathrm{s}} T_2\,]_{\leftrightarrows^{\mathrm{s}}}\,,\\
&\widetilde{a}^{\mathrm{s}} = [\,\widehat{a}^{\mathrm{s}}\,]_{\leftrightarrows^{\mathrm{s}}}\,, && [\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} \mathbin{\widetilde{\between}}^{\mathrm{s}} [\,T_2\,]_{\leftrightarrows^{\mathrm{s}}} = [\,T_1 \mathbin{\widehat{\between}}^{\mathrm{s}} T_2\,]_{\leftrightarrows^{\mathrm{s}}}\,,\\
[\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} \mathbin{\widetilde{+}}^{\mathrm{s}} [\,T_2\,]_{\leftrightarrows^{\mathrm{s}}} &= [\,T_1 \mathbin{\widehat{+}}^{\mathrm{s}} T_2\,]_{\leftrightarrows^{\mathrm{s}}}\,, && [\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} \mathbin{\widetilde{\upharpoonright}}^{\mathrm{s}} [\,T_2\,]_{\leftrightarrows^{\mathrm{s}}} = [\,T_1 \mathbin{\widehat{\upharpoonright}}^{\mathrm{s}} T_2\,]_{\leftrightarrows^{\mathrm{s}}}\,,\\
[\,T_1\,]_{\leftrightarrows^{\mathrm{s}}} \mathbin{\widetilde{\cdot}}^{\mathrm{s}} [\,T_2\,]_{\leftrightarrows^{\mathrm{s}}} &= [\,T_1 \mathbin{\widehat{\cdot}}^{\mathrm{s}} T_2\,]_{\leftrightarrows^{\mathrm{s}}}\,, && \widetilde{\partial_H}^{\mathrm{s}}([\,T_1\,]_{\leftrightarrows^{\mathrm{s}}}) = [\,\widehat{\partial_H}^{\mathrm{s}}(T_1)\,]_{\leftrightarrows^{\mathrm{s}}}\,.
\end{aligned}$$

The operations on $\mathbb{CTS}^{\epsilon\mathrm{s}}_{\kappa}/\leftrightarrows^{\mathrm{s}}$ are well-defined because $\leftrightarrows^{\mathrm{s}}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}^{\epsilon\mathrm{s}}_{\kappa}$.

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

Theorem 10.1 (Soundness of $\mathrm{ACP}^{\mathrm{cs}}_{\epsilon}$). *For each infinite cardinal κ, we have $\mathfrak{P}^{\epsilon\mathrm{cs}}_{\kappa} \models \mathrm{ACP}^{\mathrm{cs}}_{\epsilon}$.*

Proof. Because $\leftrightarrows^{\mathrm{s}}$ is a congruence, it is sufficient to show that all axioms are sound. The soundness of all axioms follows straightforwardly from the definition of $\mathfrak{P}^{\epsilon\mathrm{cs}}_{\kappa}$. □

For all axioms that are in common with $\mathrm{ACP}^{\mathrm{cs}}$, the proof of soundness with respect to $\mathfrak{P}^{\epsilon\mathrm{cs}}_{\kappa}$ follows the same line as the proof of soundness with respect to $\mathfrak{P}^{\mathrm{cs}}_{\kappa}$.

Table 14. Axioms adapted to retrospection ($a \in \mathsf{A}_\delta$)

$a \cdot x \between y = a \cdot (x \parallel \Pi^+(y))$	CM3R

Table 15. Additional axioms for retrospection ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$)

$\sim\bot = \bot$	R1	$\Pi^+_{>n}(a \cdot x) = a \cdot \Pi^+_{>n+1}(x)$	RS2
$\sim\top = \top$	R2	$\Pi^+_{>n}(x + y) = \Pi^+_{>n}(x) + \Pi^+_{>n}(y)$	RS3
$\sim(-\phi) = -(\sim\phi)$	R3	$\Pi^+_{>n}(\phi :\rightarrow x) = \Pi^+_{>n}(\phi) :\rightarrow \Pi^+_{>n}(x)$	RS4
$\sim(\phi \sqcup \psi) = \sim\phi \sqcup \sim\psi$	R4	$\Pi^+_{>n}(\bot) = \bot$	RS5
$\sim(\phi \sqcap \psi) = \sim\phi \sqcap \sim\psi$	R5	$\Pi^+_{>n}(\top) = \top$	RS6
$a \cdot (\sim\phi :\rightarrow x) =$		$\Pi^+_{>n}(\eta) = \eta$	RS7
$\phi :\rightarrow a \cdot x + -\phi :\rightarrow a \cdot \delta$	R6	$\Pi^+_{>n}(-\phi) = -\Pi^+_{>n}(\phi)$	RS8
		$\Pi^+_{>n}(\phi \sqcup \psi) = \Pi^+_{>n}(\phi) \sqcup \Pi^+_{>n}(\psi)$	RS9
		$\Pi^+_{>n}(\phi \sqcap \psi) = \Pi^+_{>n}(\phi) \sqcap \Pi^+_{>n}(\psi)$	RS10
$\Pi^+(x) = \Pi^+_{>0}(x)$	RS0	$\Pi^+_{>0}(\sim\phi) = \sim(\sim\phi)$	RS11
$\Pi^+_{>n}(\epsilon) = \epsilon$	RS1T	$\Pi^+_{>n+1}(\sim\phi) = \sim\Pi^+_{>n}(\phi)$	RS12

11 ACP$_\epsilon$ with Retrospective Conditions

In this section, we present an extension of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ with a retrospection operator on conditions. The retrospection operator allows for looking back on conditions under which preceding actions have been performed. The extension of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ with the retrospection operator is called $\mathrm{ACP}^{\mathrm{cr}}_\epsilon$.

$\mathrm{ACP}^{\mathrm{cr}}_\epsilon$ has the constants and operators of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ and in addition:

- the unary *retrospection* operator $\sim : \mathbf{C} \to \mathbf{C}$;
- the unary *retrospection shift* operator $\Pi^+ : \mathbf{P} \to \mathbf{P}$;
- for each $n \in \mathbb{N}$, the unary *restricted retrospection shift* operator $\Pi^+_{>n}{:}\mathbf{P} \to \mathbf{P}$;
- for each $n \in \mathbb{N}$, the unary *restricted retrospection shift* operator $\Pi^+_{>n}{:}\mathbf{C} \to \mathbf{C}$.

In the parallel composition of two processes, when an action of one of the processes is performed, the retrospections of the other process that are not internal should go one step further. This is accomplished by the retrospection shift operator. The restricted retrospection shift operators, on processes and conditions, are needed for the axiomatization of the retrospection shift operator. The retrospection shift operator Π^+ is similar to the history pointer shift operator *hps* from [4].

The axioms of $\mathrm{ACP}^{\mathrm{cr}}_\epsilon$ are the axioms of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ with axiom CM3 replaced by axiom CM3R from Table 14, and the additional axioms for retrospection given in Table 15. The crucial axiom is R6, which shows that a conditional expression of the form $\sim\zeta :\rightarrow p$ gives a retrospection at the condition under which the immediately preceding action has been performed. Axiom CM3R shows that retrospections are adapted if two processes proceed in parallel. Axioms RS0,

Table 16. Transition rules adapted to retrospection

$$\frac{x \xrightarrow{[\phi]\,a} x'}{x \parallel y \xrightarrow{[\phi]\,a} x' \parallel \Pi^+(y)} \qquad \frac{y \xrightarrow{[\phi]\,a} y'}{x \parallel y \xrightarrow{[\phi]\,a} \Pi^+(x) \parallel y'} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{x \lfloor\!\lfloor y \xrightarrow{[\phi]\,a} x' \parallel \Pi^+(y)}$$

RS1T and RS2–RS12 state that this happens as explained above. By means of axioms RS5–RS12, the retrospection shift operators on conditions can be eliminated from all terms of sort **C**.

Recall that we write $p \lhd \zeta \rhd q$ for $\zeta :\rightarrow p + -\zeta :\rightarrow q$. An interesting equation is $a \cdot (x \lhd {\sim}\phi \rhd y) = a \cdot x \lhd \phi \rhd a \cdot y$. This equation is a generalization of axiom R6: axiom R6 is derivable from the other axioms of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$ and this equation by substituting δ for y and applying axioms GC3 and A6. It is not immediately clear that this equation is derivable from the axioms of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$.

Proposition 11.1 (Derivability Generalization Axiom R6). *The equation* $a \cdot (x \lhd {\sim}\phi \rhd y) = a \cdot x \lhd \phi \rhd a \cdot y$ (R6′) *is derivable from the axioms of* $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$.

Proof. The proof is the same to the proof of the corresponding proposition in the setting of $\mathrm{ACP}^{\mathrm{cr}}$ given in [9]. □

Because of the addition of the retrospection operator, we cannot use the Boolean algebras $\mathcal{C}_\kappa$ here. The algebras $\mathcal{C}^{\mathrm{r}}_\kappa$ that we use here can be characterized as the free κ-complete algebras over C_{at} from the class of algebras with interpretations for the constants and operators of Boolean algebras and the retrospection operator that satisfy the axioms of Boolean algebras (Table 1) and axioms R1–R5 from Table 15. We do not make this fully precise, but give an explicit construction of the algebras $\mathcal{C}^{\mathrm{r}}_\kappa$ instead. Important to bear in mind is that not only the atomic conditions, but also the results of applying the operation associated with the retrospection operator a finite number of times to atomic conditions, should not satisfy any equations except those derivable from the axioms.

Let $\mathsf{C}^{\mathrm{r}}_{\mathsf{at}} = \bigcup\{\mathsf{C}_{\mathsf{at}} \times \{i\} \mid i \in \omega\}$ and define $\mathrm{prev} : \mathsf{C}^{\mathrm{r}}_{\mathsf{at}} \to \mathsf{C}^{\mathrm{r}}_{\mathsf{at}}$ by $\mathrm{prev}((\eta, i)) = (\eta, i+1)$. For any infinite cardinal κ, let $\mathcal{C}'_\kappa$ be the free κ-complete Boolean algebra over $\mathsf{C}^{\mathrm{r}}_{\mathsf{at}}$. Then the function prev extends to a unique κ-complete endomorphism prev^* of $\mathcal{C}'_\kappa$. This endomorphism is a unary operation on $\mathcal{C}'_\kappa$ that satisfies axioms R1–R5 from Table 15 and preserves $\bigsqcup C'$ for every $C' \subseteq \mathcal{C}'_\kappa$ of cardinality less then κ. The algebra $\mathcal{C}^{\mathrm{r}}_\kappa$ is the expansion of $\mathcal{C}'_\kappa$ obtained by associating the operation prev^* with the operator $\sim$. We write $\mathcal{C}^{\mathrm{r}}$ for $\mathcal{C}^{\mathrm{r}}_{\aleph_0}$.

The structural operational semantics of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$ is described by the transition rules for $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$ with the second and third transition rule for parallel composition and the one transition rule for left merge replaced by the transition rules given in Table 16, and the additional transition rules for retrospection given in Table 17. Of course, the conditions involved are now taken from $\mathcal{C}^{\mathrm{r}}$ instead of $\mathcal{C}$.

Table 17. Additional transition rules for retrospection

$$\frac{x\ {}^{[\phi]}\!\downarrow}{\Pi^+(x)\ {}^{[\Pi^+_{>0}(\phi)]}\!\downarrow} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\Pi^+(x) \xrightarrow{[\Pi^+_{>0}(\phi)]\,a} \Pi^+_{>1}(x')}$$

$$\frac{x\ {}^{[\phi]}\!\downarrow}{\Pi^+_{>n}(x)\ {}^{[\Pi^+_{>n}(\phi)]}\!\downarrow} \qquad \frac{x \xrightarrow{[\phi]\,a} x'}{\Pi^+_{>n}(x) \xrightarrow{[\Pi^+_{>n}(\phi)]\,a} \Pi^+_{>n+1}(x')}$$

12 Full Retrospective Splitting Bisimulation Models of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$

The construction of the full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$ differs from the construction of the full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$ in the conditions involved and in the notion of splitting bisimulation used. The conditions are now taken from $\mathcal{C}^{\mathrm{r}}_{\kappa}$ instead of $\mathcal{C}_{\kappa}$. Henceforth, we write $\mathcal{C}^{\mathrm{r}}_{\kappa}{}^{-}$ for $\mathcal{C}^{\mathrm{r}}_{\kappa} \setminus \{\bot\}$.

Let κ be an infinite cardinal. Then a *κ-conditional transition system with retrospection* T consists of the following:

- a set S of *states*;
- a set $\xrightarrow{\ell} \subseteq S \times S$, for each $\ell \in \mathcal{C}^{\mathrm{r}}_{\kappa}{}^{-} \times \mathsf{A}$;
- a set ${}^{[\alpha]}\!\downarrow \subseteq S$, for each $\alpha \in \mathcal{C}^{\mathrm{r}}_{\kappa}{}^{-}$;
- an *initial state* $s^0 \in S$.

For conditional transition systems with retrospection, reachability, connectedness and connected part are defined exactly as for conditional transition systems.

Let κ be an infinite cardinal. Then $\mathbb{CTS}^{\epsilon\mathrm{r}}_{\kappa}$ is the set of all connected κ-conditional transition systems with retrospection $T = (S, \rightarrow, \downarrow, s^0)$ such that $S \subset \mathcal{S}_{\kappa}$ and the branching degree of T is less than κ.

Isomorphism between conditional transition systems with retrospection is defined exactly as for conditional transition systems. Splitting bisimilarity has to be adapted to the setting with retrospection.

Let $T_1 = (S_1, \rightarrow_1, \downarrow_1, s^0_1) \in \mathbb{CTS}^{\epsilon\mathrm{r}}_{\kappa}$ and $T_2 = (S_2, \rightarrow_2, \downarrow_2, s^0_2) \in \mathbb{CTS}^{\epsilon\mathrm{r}}_{\kappa}$ (for an infinite cardinal κ). Then a *retrospective splitting bisimulation* B between T_1 and T_2 is a ternary relation $B \subseteq S_1 \times \mathcal{C}^{\mathrm{r}}_{\kappa} \times S_2$ such that $B(s^0_1, \top, s^0_2)$ and for all s_1, β, s_2 such that $B(s_1, \beta, s_2)$:

- if $s_1 \xrightarrow{[\alpha]\,a}_1 s'_1$, then there is a set $CS'_2 \subseteq \mathcal{C}^{\mathrm{r}}_{\kappa}{}^{-} \times S_2$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup \mathrm{dom}(CS'_2)$ and for all $(\alpha', s'_2) \in CS'_2$, $s_2 \xrightarrow{[\alpha']\,a}_2 s'_2$ and $B(s'_1, {\sim}\alpha', s'_2)$;
- if $s_2 \xrightarrow{[\alpha]\,a}_2 s'_2$, then there is a set $CS'_1 \subseteq \mathcal{C}^{\mathrm{r}}_{\kappa}{}^{-} \times S_1$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup \mathrm{dom}(CS'_1)$ and for all $(\alpha', s'_1) \in CS'_1$, $s_1 \xrightarrow{[\alpha']\,a}_1 s'_1$ and $B(s'_1, {\sim}\alpha', s'_2)$;

– if $s_1 \stackrel{[\alpha]}{\downarrow}_1$, then there is a set $C' \subseteq \mathcal{C}^{\mathrm{r}-}_\kappa$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \stackrel{[\alpha']}{\downarrow}_2$;
– if $s_2 \stackrel{[\alpha]}{\downarrow}_2$, then there is a set $C' \subseteq \mathcal{C}^{\mathrm{r}-}_\kappa$ of cardinality less than κ such that $\alpha \sqcap \beta \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \stackrel{[\alpha']}{\downarrow}_1$.

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

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

The elements of $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ and operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ to be associated with the constants and operators of $\mathrm{ACP}^{\mathrm{c}}_\epsilon$ are defined exactly as the elements of $\mathbb{CTS}^{\epsilon}_\kappa$ and operations on $\mathbb{CTS}^{\epsilon}_\kappa$ associated with them, except for $\|$, $\lfloor\!\lfloor$ and $|$. The operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ that we associate with $\|$, $\lfloor\!\lfloor$, $|$, Π^+ and $\Pi^+_{>n}$ call for unfolding of transition systems from $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$.

For the sake of unfolding, it is assumed that, for each infinite cardinal κ, $\mathcal{S}_\kappa$ has the following closure property:[9]

$$\text{for all } S \subseteq \mathcal{S}_\kappa,\ \{\pi \frown \langle s\rangle \mid \pi \in (S \times (\mathcal{C}^{\mathrm{r}}_\kappa \times \mathsf{A}))^*, s \in S\} \subseteq \mathcal{S}_\kappa\ .$$

We write $\mathrm{P}'(S)$ for the set $\{\pi \frown \langle s\rangle \mid \pi \in (S \times (\mathcal{C}^{\mathrm{r}}_\kappa \times \mathsf{A}))^*, s \in S\}$. The function $\# : \mathrm{P}'(S) \to \mathbb{N}$ is defined by

$$\begin{aligned} \#(\langle s\rangle) &= 0\ ,\\ \#(\pi \frown \langle s, \ell, s'\rangle) &= \#(\pi \frown \langle s\rangle) + 1\ . \end{aligned}$$

The elements of $\mathrm{P}'(S)$, for an $S \subseteq \mathcal{S}_\kappa$, can be looked upon as potential paths of a κ-conditional transition system with S as set of states. A path of a transition system $(S, \rightarrow, \downarrow, s^0) \in \mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ is a finite alternating sequence $\langle s_0, \ell_1, s_1, \ldots, \ell_n, s_n\rangle$ of states from S and labels from $\mathcal{C}^{\mathrm{r}}_\kappa \times \mathsf{A}$ such that $s_0 = s^0$ and $s_i \xrightarrow{\ell_{i+1}} s_{i+1}$ for all $i < n$. The state s_n is called the state in which the path ends.

Let $T = (S, \rightarrow, \downarrow, s^0) \in \mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$. Then the set of *paths* of T, written $\mathrm{P}(T)$, is the smallest subset of $\mathrm{P}'(S)$ such that:

– $\langle s^0\rangle \in \mathrm{P}(T)$,
– if $\pi \frown \langle s\rangle \in \mathrm{P}(T)$ and $s \xrightarrow{\ell} s'$, then $\pi \frown \langle s, \ell, s'\rangle \in \mathrm{P}(T)$.

In order to unfold a transition system, we need for each state s of the original transition system, for each different path that ends in state s, a different state in

[9] We write $\langle\,\rangle$ for the empty sequence, $\langle e\rangle$ for the sequence having e as sole element and $\sigma \frown \sigma'$ for the concatenation of sequences σ and σ'; and we use $\langle e_1, \ldots, e_n\rangle$ as a shorthand for $\langle e_1\rangle \frown \ldots \frown \langle e_n\rangle$.

the unfolded transition system. The obvious choice is to take the paths concerned as states.

Let $T = (S, \rightarrow, \downarrow, s^0) \in \mathbb{CTS}^{\mathrm{er}}_\kappa$. Then the *unfolding* of T, written $\Upsilon(T)$, is defined as follows:

$$\Upsilon(T) = (S', \rightarrow', \downarrow', s^{0\prime}) \ ,$$

where

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

and for every $\ell \in \mathcal{C}^{\mathrm{r}-}_\kappa \times \mathsf{A}$ and $\alpha \in \mathcal{C}^{\mathrm{r}-}_\kappa$:

$$\begin{aligned}
&\xrightarrow{\ell}{}' = \{(\pi \frown \langle s \rangle, \pi \frown \langle s, \ell, s' \rangle) \mid \pi \frown \langle s \rangle \in \mathrm{P}(T), s \xrightarrow{\ell} s'\} \ , \\
&{}^{[\alpha]}\!\downarrow' = \{\pi \frown \langle s \rangle \mid \pi \frown \langle s \rangle \in \mathrm{P}(T), s \ {}^{[\alpha]}\!\downarrow\} \ , \\
&s^{0\prime} \ = \langle s^0 \rangle \ .
\end{aligned}$$

The functions upd_1 and upd_2 defined next will be used in the definition of parallel composition on $\mathbb{CTS}^{\mathrm{er}}_\kappa$ to adapt the retrospection in steps originating from the first operand and the second operand, respectively.

Let $S_1, S_2 \subseteq \mathcal{S}_\kappa$. Then the functions $\mathrm{upd}_i : \mathcal{C}^{\mathrm{r}-}_\kappa \times \mathrm{P}'(S_1 \times S_2) \to \mathcal{C}^{\mathrm{r}-}_\kappa$, for $i = 1, 2$, are defined by

$$\begin{aligned}
&\mathrm{upd}_i(\alpha, \langle (s_1, s_2) \rangle) &&= \alpha \ , \\
&\mathrm{upd}_i(\alpha, \langle (s_1, s_2), \ell, (s_1', s_2') \rangle \frown \pi') &&= \mathrm{upd}_i(\alpha, \langle (s_1', s_2') \rangle \frown \pi') \text{ if } s_i \neq s_i' \ , \\
&\mathrm{upd}_i(\alpha, \langle (s_1, s_2), \ell, (s_1', s_2') \rangle \frown \pi') &&= \\
&\qquad \mathrm{upd}_i(\Pi^+_{>\#_i(\langle (s_1', s_2') \rangle \frown \pi')}(\alpha), \langle (s_1', s_2') \rangle \frown \pi') &&\text{ if } s_i = s_i' \ .
\end{aligned}$$

where

$$\begin{aligned}
&\#_i(\langle (s_1, s_2) \rangle) &&= 0 \ , \\
&\#_i(\langle (s_1, s_2), \ell, (s_1', s_2') \rangle \frown \pi') &&= \#_i(\langle (s_1', s_2') \rangle \frown \pi') + 1 \text{ if } s_i \neq s_i' \ , \\
&\#_i(\langle (s_1, s_2), \ell, (s_1', s_2') \rangle \frown \pi') &&= \#_i(\langle (s_1', s_2') \rangle \frown \pi') \qquad \text{if } s_i = s_i' \ .
\end{aligned}$$

Henceforth, we write $\mathrm{upd}(\alpha_1, \alpha_2, \pi)$ for $\mathrm{upd}_1(\alpha_1, \pi) \sqcap \mathrm{upd}_2(\alpha_2, \pi)$.

We proceed with associating operations on $\mathbb{CTS}^{\mathrm{er}}_\kappa$ with the operators $\|$, $\lfloor\!\lfloor$, $|$, Π^+ and $\Pi^+_{>n}$.

We associate with the additional operator $\|$ an operation $\mathbin{\widehat{\|}^{\mathrm{r}}}$ on $\mathbb{CTS}^{\mathrm{er}}_\kappa$ as follows.

– Let $T_1, T_2 \in \mathbb{CTS}^{\mathrm{er}}_\kappa$. Suppose that $\Upsilon(T_i) = (S_i, \rightarrow_i, \downarrow_i, s_i^0)$ for $i = 1, 2$, and $\Upsilon(\Upsilon(T_1) \mathbin{\widehat{\|}} \Upsilon(T_2)) = (S, \rightarrow, \downarrow, s^0)$. Then

$$T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2 = (S, \rightarrow', \downarrow', s^0) \ ,$$

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

$$
\begin{aligned}
\xrightarrow{(\alpha,a)}{}' = {} & \{(\pi \frown \langle (s_1,s_2)\rangle, \pi' \frown \langle (s_1',s_2')\rangle) \mid s_1 \neq s_1' \ \wedge \ s_2 = s_2' \ \wedge \\
& \qquad \bigvee_{\alpha' \in \mathcal{C}^{\mathrm{r}}_\kappa -} (\pi \frown \langle (s_1,s_2)\rangle \xrightarrow{[\alpha']\, a} \pi' \frown \langle (s_1',s_2')\rangle \ \wedge \\
& \qquad\qquad \mathrm{upd}_1(\alpha', \pi \frown \langle (s_1,s_2)\rangle) = \alpha)\} \\
& \cup \{(\pi \frown \langle (s_1,s_2)\rangle, \pi' \frown \langle (s_1',s_2')\rangle) \mid s_1 = s_1' \ \wedge \ s_2 \neq s_2' \ \wedge \\
& \qquad \bigvee_{\alpha' \in \mathcal{C}^{\mathrm{r}}_\kappa -} (\pi \frown \langle (s_1,s_2)\rangle \xrightarrow{[\alpha']\, a} \pi' \frown \langle (s_1',s_2')\rangle \ \wedge \\
& \qquad\qquad \mathrm{upd}_2(\alpha', \pi \frown \langle (s_1,s_2)\rangle) = \alpha)\} \\
& \cup \{(\pi \frown \langle (s_1,s_2)\rangle, \pi' \frown \langle (s_1',s_2')\rangle) \mid \\
& \qquad \bigvee_{\alpha',\beta' \in \mathcal{C}^{\mathrm{r}}_\kappa -, a', b' \in \mathsf{A}} (\pi \frown \langle (s_1,s_2)\rangle \xrightarrow{[\alpha' \sqcap \beta']\, a} \pi' \frown \langle (s_1',s_2')\rangle \ \wedge \\
& \qquad\qquad s_1 \xrightarrow{[\alpha']\, a'}_1 s_1' \ \wedge \ s_2 \xrightarrow{[\beta']\, b'}_2 s_2' \ \wedge \\
& \qquad\qquad \mathrm{upd}(\alpha', \beta', \pi \frown \langle (s_1,s_2)\rangle) = \alpha \ \wedge \\
& \qquad\qquad\qquad a' \mid b' = a)\} \ , \\
{}^{[\alpha'']}\!\downarrow' = {} & \{\pi \frown \langle (s_1,s_2)\rangle \mid \\
& \qquad \bigvee_{\alpha',\beta' \in \mathcal{C}^{\mathrm{r}}_\kappa -} (\pi \frown \langle (s_1,s_2)\rangle \ {}^{[\alpha' \sqcap \beta']}\!\downarrow \wedge \ s_1 \ {}^{[\alpha']}\!\downarrow_1 \wedge \ s_2 \ {}^{[\beta']}\!\downarrow_2 \wedge \\
& \qquad\qquad \mathrm{upd}(\alpha', \beta', \pi \frown \langle (s_1,s_2)\rangle) = \alpha'')\} \ .
\end{aligned}
$$

Remark 12.1. The operation $\widehat{\|}^{\mathrm{r}}$ on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ is defined above in a step-by-step way. The basic idea behind this definition is twofold:

- $T_1 \mathbin{\widehat{\|}^{\mathrm{r}}} T_2$ can be obtained by first composing T_1 and T_2 to $T_1 \mathbin{\widehat{\|}} T_2$ and then adapting the retrospections in steps of $T_1 \mathbin{\widehat{\|}} T_2$;
- unfolding of $T_1 \mathbin{\widehat{\|}} T_2$ is needed before the actual adaptations can take place because the adaptation of the retrospection in a step may be different for the different paths that end in the state from which the step starts.

Somewhat surprisingly, in addition, T_1 and T_2 must be unfolded before the actual composition takes place. In a step where an action of T_1 and an action of T_2 are performed synchronously, the condition under which the action of T_1 can be performed and the condition under which the action of T_2 can be performed are needed to adapt the retrospection in that step correctly. If T_1 and T_2 are not unfolded before the actual composition takes place, in general, those conditions cannot be determined uniquely.

The operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ to be associated with the additional operators $\mathbin{\|}$ and $\mid$ are defined analogously. The operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ to be associated with the additional operators ∂_H are defined exactly as the operations on $\mathbb{CTS}^{\epsilon}_\kappa$ associated with them. We associate with the additional operators $\Pi^+_{>n}$ operations $\widehat{\Pi^+_{>n}}^{\,\mathrm{r}}$ on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ as follows.

- Let $T \in \mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$. Suppose that $\Upsilon(T) = (S, \rightarrow, \downarrow, s^0)$. Then

$$\widehat{\Pi^+_{>n}}^{\,\mathrm{r}}(T) = (S, \rightarrow', \downarrow', s^0) \ ,$$

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

$$\begin{aligned}
\xrightarrow{(\alpha,a)}{}' &= \{(\pi \frown \langle s\rangle, \pi' \frown \langle s'\rangle) \mid \\
&\qquad \bigvee_{\alpha' \in {\mathcal{C}^{\mathrm{r}}_\kappa}^-} (\pi \frown \langle s\rangle \xrightarrow{[\alpha']\, a} \pi' \frown \langle s'\rangle \wedge \Pi^+_{>\#(\pi)+n}(\alpha') = \alpha)\} \,, \\
{}^{[\alpha'']}\!\downarrow' &= \{\pi \frown \langle s\rangle \mid \bigvee_{\alpha' \in {\mathcal{C}^{\mathrm{r}}_\kappa}^-} (\pi \frown \langle s\rangle \, {}^{[\alpha']}\!\downarrow \wedge \Pi^+_{>\#(\pi)+n}(\alpha') = \alpha'')\} \,.
\end{aligned}$$

The operation on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ to be associated with the additional operator Π^+ is the same as the operation on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ associated with $\Pi^+_{>0}$.

We can show that retrospective splitting bisimilarity is a congruence with respect to the operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ associated with the operators of $\mathrm{ACP}^{\mathrm{cr}}_\epsilon$.

Proposition 12.1 (Congruence). *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$ and $\alpha \in \mathcal{C}_\kappa$, $T_1 \leftrightarrows^{\mathrm{r}} T_1'$ and $T_2 \leftrightarrows^{\mathrm{r}} T_2'$ imply $T_1 \mathbin{\widehat{+}}^{\mathrm{r}} T_2 \leftrightarrows^{\mathrm{r}} T_1' \mathbin{\widehat{+}}^{\mathrm{r}} T_2'$, $T_1 \mathbin{\widehat{\cdot}}^{\mathrm{r}} T_2 \leftrightarrows^{\mathrm{r}} T_1' \mathbin{\widehat{\cdot}}^{\mathrm{r}} T_2'$, $\alpha \mathbin{\widehat{\rightarrow}}^{\mathrm{r}} T_1 \leftrightarrows^{\mathrm{r}} \alpha \mathbin{\widehat{\rightarrow}}^{\mathrm{r}} T_1'$, $T_1 \mathbin{\widehat{\parallel}}^{\mathrm{r}} T_2 \leftrightarrows^{\mathrm{r}} T_1' \mathbin{\widehat{\parallel}}^{\mathrm{r}} T_2'$, $T_1 \mathbin{\widehat{\between}}^{\mathrm{r}} T_2 \leftrightarrows^{\mathrm{r}} T_1' \mathbin{\widehat{\between}}^{\mathrm{r}} T_2'$, $T_1 \mathbin{\widehat{\lhd}}^{\mathrm{r}} T_2 \leftrightarrows^{\mathrm{r}} T_1' \mathbin{\widehat{\lhd}}^{\mathrm{r}} T_2'$, $\widehat{\partial_H}^{\mathrm{r}}(T_1) \leftrightarrows^{\mathrm{r}} \widehat{\partial_H}^{\mathrm{r}}(T_1')$, $\widehat{\Pi^+}^{\mathrm{r}}(T_1) \leftrightarrows^{\mathrm{r}} \widehat{\Pi^+}^{\mathrm{r}}(T_1')$ and $\widehat{\Pi^+_{>n}}^{\mathrm{r}}(T_1) \leftrightarrows^{\mathrm{r}} \widehat{\Pi^+_{>n}}^{\mathrm{r}}(T_1')$.*

Proof. For all operations, witnessing splitting bisimulations are constructed in the same way as in the congruence proofs for the corresponding operations on $\mathbb{CTS}^{\mathrm{r}}_\kappa$ given in [9]. □

The ingredients of the *full retrospective splitting bisimulation models* $\mathfrak{P}^{\epsilon\mathrm{cr}}_\kappa$ of $\mathrm{ACP}^{\mathrm{cr}}_\epsilon$, one for each infinite cardinal κ, are defined as follows:

$$\begin{aligned}
\mathcal{P} &= \mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa / \leftrightarrows^{\mathrm{r}} \,, \\
\widetilde{\delta}^{\mathrm{r}} &= [\,\widehat{\delta}^{\mathrm{r}}\,]_{\leftrightarrows^{\mathrm{r}}} \,, & [T_1]_{\leftrightarrows^{\mathrm{r}}} \mathbin{\widetilde{\parallel}}^{\mathrm{r}} [T_2]_{\leftrightarrows^{\mathrm{r}}} &= [T_1 \mathbin{\widehat{\parallel}}^{\mathrm{r}} T_2]_{\leftrightarrows^{\mathrm{r}}} \,, \\
\widetilde{\epsilon}^{\mathrm{r}} &= [\,\widehat{\epsilon}^{\mathrm{r}}\,]_{\leftrightarrows^{\mathrm{r}}} \,, & [T_1]_{\leftrightarrows^{\mathrm{r}}} \mathbin{\widetilde{\between}}^{\mathrm{r}} [T_2]_{\leftrightarrows^{\mathrm{r}}} &= [T_1 \mathbin{\widehat{\between}}^{\mathrm{r}} T_2]_{\leftrightarrows^{\mathrm{r}}} \,, \\
\widetilde{a}^{\mathrm{r}} &= [\,\widehat{a}^{\mathrm{r}}\,]_{\leftrightarrows^{\mathrm{r}}} \,, & [T_1]_{\leftrightarrows^{\mathrm{r}}} \mathbin{\widetilde{\lhd}}^{\mathrm{r}} [T_2]_{\leftrightarrows^{\mathrm{r}}} &= [T_1 \mathbin{\widehat{\lhd}}^{\mathrm{r}} T_2]_{\leftrightarrows^{\mathrm{r}}} \,, \\
[T_1]_{\leftrightarrows^{\mathrm{r}}} \mathbin{\widetilde{+}}^{\mathrm{r}} [T_2]_{\leftrightarrows^{\mathrm{r}}} &= [T_1 \mathbin{\widehat{+}}^{\mathrm{r}} T_2]_{\leftrightarrows^{\mathrm{r}}} \,, & \widetilde{\partial_H}^{\mathrm{r}}([T_1]_{\leftrightarrows^{\mathrm{r}}}) &= [\widehat{\partial_H}^{\mathrm{r}}(T_1)]_{\leftrightarrows^{\mathrm{r}}} \,, \\
[T_1]_{\leftrightarrows^{\mathrm{r}}} \mathbin{\widetilde{\cdot}}^{\mathrm{r}} [T_2]_{\leftrightarrows^{\mathrm{r}}} &= [T_1 \mathbin{\widehat{\cdot}}^{\mathrm{r}} T_2]_{\leftrightarrows^{\mathrm{r}}} \,, & \widetilde{\Pi^+}^{\mathrm{r}}([T_1]_{\leftrightarrows^{\mathrm{r}}}) &= [\widehat{\Pi^+}^{\mathrm{r}}(T_1)]_{\leftrightarrows^{\mathrm{r}}} \,, \\
\alpha \mathbin{\widetilde{\rightarrow}}^{\mathrm{r}} [T_1]_{\leftrightarrows^{\mathrm{r}}} &= [\,\alpha \mathbin{\widehat{\rightarrow}}^{\mathrm{r}} T_1\,]_{\leftrightarrows^{\mathrm{r}}} \,, & \widetilde{\Pi^+_{>n}}^{\mathrm{r}}([T_1]_{\leftrightarrows^{\mathrm{r}}}) &= [\widehat{\Pi^+_{>n}}^{\mathrm{r}}(T_1)]_{\leftrightarrows^{\mathrm{r}}} \,.
\end{aligned}$$

The operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa / \leftrightarrows^{\mathrm{r}}$ are well-defined because $\leftrightarrows^{\mathrm{r}}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}^{\epsilon\mathrm{r}}_\kappa$.

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

Theorem 12.1 (Soundness of $\mathrm{ACP}^{\mathrm{cr}}_\epsilon$). *For each infinite cardinal κ, we have $\mathfrak{P}^{\epsilon\mathrm{cr}}_\kappa \models \mathrm{ACP}^{\mathrm{cr}}_\epsilon$.*

Proof. Because $\leftrightarrows^{\mathrm{r}}$ is a congruence, it is sufficient to show that all axioms are sound. The soundness of all axioms follows straightforwardly from the definition of $\mathfrak{P}^{\epsilon\mathrm{cr}}_\kappa$. □

For all axioms that are in common with $\mathrm{ACP}^{\mathrm{cr}}$, the proof of soundness with respect to $\mathfrak{P}^{\epsilon\mathrm{cr}}_{\kappa}$ follows the same line as the proof of soundness with respect to $\mathfrak{P}^{\mathrm{cr}}_{\kappa}$.

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

Theorem 12.2 (Unique solutions in $\mathfrak{P}^{\epsilon\mathrm{cr}}_{\kappa}$). *For each infinite cardinal κ, guarded recursive specifications over $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$ have unique solutions in $\mathfrak{P}^{\epsilon\mathrm{cr}}_{\kappa}$.*

Proof. The proof is analogous to the proof of the corresponding property for the full retrospective splitting bisimulation models of $\mathrm{ACP}^{\mathrm{cr}}$ given in [9]. □

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

13 Evaluation of Retrospective Conditions

In this section, we add condition evaluation operators and generalized condition evaluation operators to $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$. As in the case of $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$, these operators require to fix an infinite cardinal λ. By doing so, full retrospective splitting bisimulation models with domain $\mathbb{CTS}^{\mathrm{er}}_{\kappa}/\leftrightarrows^{\mathrm{r}}$ for $\kappa > \lambda$ are excluded.

Henceforth, we write $\mathcal{H}^{\mathrm{r}}_{\lambda}$ for the set of all λ-complete endomorphisms of $\mathcal{C}^{\mathrm{r}}_{\lambda}$.

In the case of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$, there are λ-complete condition evaluation operators $\mathsf{CE}_h : \mathbf{P} \to \mathbf{P}$ and $\mathsf{CE}_h : \mathbf{C} \to \mathbf{C}$, and generalized λ-complete condition evaluation operators $\mathsf{GCE}_h : \mathbf{P} \to \mathbf{P}$ and $\mathsf{GCE}_h : \mathbf{C} \to \mathbf{C}$, for each $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$. We also need the following auxiliary operators:

- for each $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$, $n \in \mathbb{N}$, the unary *retrospection update* operator $\Pi^h_n : \mathbf{P} \to \mathbf{P}$;
- for each $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$, $n \in \mathbb{N}$, the unary *retrospection update* operator $\Pi^h_n : \mathbf{C} \to \mathbf{C}$.

In the case of $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$, it is assumed that a fixed but arbitrary function $\mathsf{eff} : \mathsf{A} \times \mathcal{H}^{\mathrm{r}}_{\lambda} \to \mathcal{H}^{\mathrm{r}}_{\lambda}$ has been given. The function eff is extended to A_{δ} such that $\mathsf{eff}(\delta, h) = h$ for all $h \in \mathcal{H}^{\mathrm{r}}_{\lambda}$.

The condition evaluation operators and generalized condition evaluation operators cannot be added to $\mathrm{ACP}^{\mathrm{cr}}_{\epsilon}$ in the same way as they are added to $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$. First of all, retrospective conditions may refer back too far to be evaluated. The effect is that, in condition evaluation or generalized condition evaluation of a process according to some endomorphism, the retrospective conditions that refer back further than the beginning of the process have to be left unevaluated. This is accomplished by the retrospection update operators mentioned above. In the case of generalized condition evaluation, there is another complication. Recall that generalized condition evaluation allows the results of condition evaluation to change by performing an action. In the presence of retrospection, different parts of a condition may have to be evaluated differently because of such changes.

Table 18. New axioms for (generalized) condition evaluation ($a \in \mathsf{A}_\delta$)

$\mathsf{CE}_h(\epsilon) = \epsilon$	CE1T
$\mathsf{CE}_h(a \cdot x) = a \cdot \mathsf{CE}_h(\Pi_1^h(x))$	CE2R
$\mathsf{CE}_h(x + y) = \mathsf{CE}_h(x) + \mathsf{CE}_h(y)$	CE3
$\mathsf{CE}_h(\phi :\rightarrow x) = \Pi_0^h(\phi) :\rightarrow \mathsf{CE}_h(x)$	CE4R
$\mathsf{GCE}_h(\epsilon) = \epsilon$	GCE1T
$\mathsf{GCE}_h(a \cdot x) = a \cdot \mathsf{GCE}_{\mathsf{eff}(a,h)}(\Pi_1^h(x))$	GCE2R
$\mathsf{GCE}_h(x + y) = \mathsf{GCE}_h(x) + \mathsf{GCE}_h(y)$	GCE3
$\mathsf{GCE}_h(\phi :\rightarrow x) = \Pi_0^h(\phi) :\rightarrow \mathsf{GCE}_h(x)$	GCE4R

Table 19. Axioms for retrospection update ($a \in \mathsf{A}_\delta$, $\eta \in \mathsf{C}_{\mathsf{at}}$, $\eta' \in \mathsf{C}_{\mathsf{at}} \cup \{\bot, \top\}$)

$\Pi_n^h(\epsilon) = \epsilon$	RU1T	$\Pi_0^h(\eta) = \eta'$	if $h(\eta) = \eta'$	RU7
$\Pi_n^h(a \cdot x) = a \cdot \Pi_{n+1}^h(x)$	RU2	$\Pi_{n+1}^h(\eta) = \eta$		RU8
$\Pi_n^h(x + y) = \Pi_n^h(x) + \Pi_n^h(y)$	RU3	$\Pi_n^h(-\phi) = -\Pi_n^h(\phi)$		RU9
$\Pi_n^h(\phi :\rightarrow x) = \Pi_n^h(\phi) :\rightarrow \Pi_n^h(x)$	RU4	$\Pi_n^h(\phi \sqcup \psi) = \Pi_n^h(\phi) \sqcup \Pi_n^h(\psi)$		RU10
		$\Pi_n^h(\phi \sqcap \psi) = \Pi_n^h(\phi) \sqcap \Pi_n^h(\psi)$		RU11
$\Pi_n^h(\bot) = \bot$	RU5	$\Pi_0^h(\sim\phi) = \sim\phi$		RU12
$\Pi_n^h(\top) = \top$	RU6	$\Pi_{n+1}^h(\sim\phi) = \sim\Pi_n^h(\phi)$		RU13

The effect is that, in generalized condition evaluation of a process according to some endomorphism, after an action of the process is performed, the subsequent retrospective conditions that refer back to the beginning of the process have to be evaluated according to that endomorphism as well. This is also accomplished by the retrospection update operators mentioned above.

In the case of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$, the additional axioms for CE_h and GCE_h, where $h \in \mathcal{H}_\lambda^{\mathrm{r}}$, are the axioms given in Tables 18 and 19. These additional axioms differ from the additional axioms in the absence of retrospection (Tables 5 and 7) in that axioms CE2, CE4, GCE2 and GCE4 have been replaced by axioms CE2R, CE4R, GCE2R and GCE4R, and axioms CE6–CE11 by axioms RU1T and RU2–RU13. Axioms CE2R and CE4R, together with axioms RU1T and RU2–RU13, show that, in condition evaluation of a process, retrospective conditions that refer back further than the beginning of the process are not at all evaluated. Similarly, axioms GCE2R and GCE4R, together with axioms RU1T and RU2–RU13, show that, in generalized condition evaluation of a process, retrospective conditions that refer back further than the beginning of the process are not at all evaluated. Moreover, axiom GCE2R, together with axioms RU1T and RU2–RU13, shows that, in generalized condition evaluation of a process according to some endomorphism, after an action of the process is performed, the subsequent retrospective conditions that refer back to the beginning of the process are evaluated according to that endomorphism as well.

Table 20. New transition rules for (generalized) condition evaluation

$$\frac{x \overset{[\phi]}{\downarrow}}{\mathsf{CE}_h(x) \overset{[\Pi_0^h(\phi)]}{\downarrow}} \Pi_0^h(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{\mathsf{CE}_h(x) \xrightarrow{[\Pi_0^h(\phi)]\, a} \mathsf{CE}_h(\Pi_1^h(x'))} \Pi_0^h(\phi) \neq \bot$$

$$\frac{x \overset{[\phi]}{\downarrow}}{\mathsf{GCE}_h(x) \overset{[\Pi_0^h(\phi)]}{\downarrow}} \Pi_0^h(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{\mathsf{GCE}_h(x) \xrightarrow{[\Pi_0^h(\phi)]\, a} \mathsf{GCE}_{\mathsf{eff}(a,h)}(\Pi_1^h(x'))} \Pi_0^h(\phi) \neq \bot$$

$$\frac{x \overset{[\phi]}{\downarrow}}{\Pi_n^h(x) \overset{[\Pi_n^h(\phi)]}{\downarrow}} \Pi_n^h(\phi) \neq \bot \qquad \frac{x \xrightarrow{[\phi]\, a} x'}{\Pi_n^h(x) \xrightarrow{[\Pi_n^h(\phi)]\, a} \Pi_{n+1}^h(x')} \Pi_n^h(\phi) \neq \bot$$

The structural operational semantics of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ extended with condition evaluation and generalized condition evaluation is described by the transition rules for $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ and the transition rules given in Table 20.

The full retrospective splitting bisimulation models of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ with condition evaluation and/or generalized condition evaluation are not simply the expansions of the full retrospective splitting bisimulation models $\mathfrak{P}_\kappa^{\epsilon\mathrm{cr}}$ of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$, for infinite cardinals $\kappa \leq \lambda$, obtained by associating with each operator CE_h and/or GCE_h the corresponding re-labeling operation on conditional transition systems with retrospection. As suggested by the structural operational semantics of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ extended with condition evaluation and generalized condition evaluation, these re-labeling operations have to be adapted in a way similar to the way in which parallel composition had to be adapted to the case with retrospection in Section 12. As mentioned before, full retrospective splitting bisimulation models with domain $\mathbb{CTS}_\kappa^{\epsilon\mathrm{r}}/\leftrightarrows^{\mathrm{r}}$ for $\kappa > \lambda$ are excluded.

Proposition 7.2, stating that the generalized λ-complete condition evaluation operators supersede the λ-complete condition evaluation operators in the setting of $\mathrm{ACP}_\epsilon^{\mathrm{c}}$, goes through in the setting of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$.

Adding state operators to $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ can be done on the same lines as adding generalized evaluation operators to $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$, but is more complicated. Roughly speaking, signal emission can be added to $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ in the same way as it is added to $\mathrm{ACP}_\epsilon^{\mathrm{c}}$ provided that signals are taken from $\mathcal{C}$. No adaptations like for generalized condition evaluation are needed because signal emission corresponds to condition evaluation that does not persist over performing an action. This property also points at one of the differences between the signal-emission approach to condition evaluation and the other approaches treated in this paper: retrospection has to be resolved in the signal-emission approach before condition evaluation can take place. The case where signals are taken from $\mathcal{C}^{\mathrm{r}}$ is expected to be too complicated to handle.

14 An Application of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$

The ultimate applications of a process algebra that includes conditional expressions of some form are the ones that remain entirely within the domain of process

Table 21. Additional axioms for last action conditions ($a \in \mathsf{A}$)

$a \cdot x = a \cdot (\mathcal{J}_a :\rightarrow x)$	J

Table 22. Axioms adapted to last action conditions ($a, b \in \mathsf{A}_\delta$, $c \in \mathsf{A}$)

$a \cdot x \mid b \cdot y = (a \mid b) \cdot (\Pi^a_0(x) \parallel \Pi^b_0(y))$	CM7J	$\Pi^a_n(\bot) = \bot$		LAU5
		$\Pi^a_n(\top) = \top$		LAU6
$\Pi^+_{>0}(\mathcal{J}_c) = {\sim}\mathcal{J}_c$	RS7Ja	$\Pi^a_0(\mathcal{J}_c) = \bot$	if $a \neq c$	LAU7
$\Pi^+_{>n+1}(\mathcal{J}_c) = \mathcal{J}_c$	RS7Jb	$\Pi^a_0(\mathcal{J}_c) = \top$	if $a = c$	LAU8
		$\Pi^a_{n+1}(\mathcal{J}_c) = \mathcal{J}_c$		LAU9
		$\Pi^a_n(-\phi) = -\Pi^a_n(\phi)$		LAU10
$\Pi^a_n(\epsilon) = \epsilon$	LAU1T	$\Pi^a_n(\phi \sqcup \psi) = \Pi^a_n(\phi) \sqcup \Pi^a_n(\psi)$		LAU11
$\Pi^a_n(b \cdot x) = b \cdot \Pi^a_{n+1}(x)$	LAU2	$\Pi^a_n(\phi \sqcap \psi) = \Pi^a_n(\phi) \sqcap \Pi^a_n(\psi)$		LAU12
$\Pi^a_n(x + y) = \Pi^a_n(x) + \Pi^a_n(y)$	LAU3	$\Pi^a_0({\sim}\phi) = {\sim}\phi$		LAU13
$\Pi^a_n(\phi :\rightarrow x) = \Pi^a_n(\phi) :\rightarrow \Pi^a_n(x)$	LAU4	$\Pi^a_{n+1}({\sim}\phi) = {\sim}\Pi^a_n(\phi)$		LAU14

algebra. Such applications are by their nature extensions as well. We outline one interesting application of this kind in the setting of $\mathrm{ACP}^{\mathrm{cr}}_\epsilon$.

We take the set $\{\mathcal{J}_a \mid a \in \mathsf{A}\}$ of *last action conditions* as the set of atomic conditions C_{at}. The intuition is that $\mathcal{J}_a$ indicates that action a is performed just now. The retrospection operator now allows for using conditions which express that a certain number of steps ago a certain action must have been performed.

Because we remain entirely within the domain of process algebra some additional axioms are needed. They are given in Table 21. Moreover, axioms CM7 (Table 1) and RS7 (Table 15) must be replaced by axioms CM7J and RS7Ja–RS7Jb from Table 22. Axiom CM7 must be replaced by axiom CM7J because, after performing $a \mid b$, it makes no sense to refer back to the actions performed just now by the processes originally following a and b in the process following $a \mid b$. Retrospective conditions in the process originally following a that indicate that a is performed just now should be evaluated to $\top$ and the ones that indicate that another action is performed just now should be evaluated to $\bot$. Retrospective conditions in the process originally following b should be evaluated analogously. This is accomplished by the auxiliary operators $\Pi^a_n : \mathbf{P} \to \mathbf{P}$ and $\Pi^a_n : \mathbf{C} \to \mathbf{C}$ (for each $a \in \mathsf{A}_\delta$ and $n \in \mathbb{N}$) of which the defining axioms are LAU1T and LAU2–LAU14 from Table 22. Axiom RS7 must be replaced by axioms RS7Ja and RS7Jb because of the retrospective nature of last action conditions. We mean by this that $\mathcal{J}_a$ can be viewed as a condition of the form ${\sim}\eta$, where η indicates that action a is performed next. We have not introduced corresponding atomic conditions because their use without restrictions would be problematic in alternative composition.

From the axioms of $\mathrm{BPA}^{\mathrm{cr}}_{\delta\epsilon}$ and the additional axiom J, we can derive the equation $a{\cdot}x{+}b{\cdot}y = (a{+}b){\cdot}(\mathcal{J}_a{:\rightarrow}x{+}\mathcal{J}_b{:\rightarrow}y)$. It can be used to reduce the number

of subprocesses of a process. For example, $a \cdot (a_1 \cdot a'_1 + a_2 \cdot a'_2) + b \cdot (b_1 \cdot b'_1 + b_2 \cdot b'_2) = (a+b) \cdot (\mathcal{J}_a{:}{\rightarrow}(a_1+a_2) \cdot (\mathcal{J}_{a_1}{:}{\rightarrow}a'_1 + \mathcal{J}_{a_2}{:}{\rightarrow}a'_2) + \mathcal{J}_b{:}{\rightarrow}(b_1+b_2) \cdot (\mathcal{J}_{b_1}{:}{\rightarrow}b'_1 + \mathcal{J}_{b_2}{:}{\rightarrow}b'_2))$ shows a reduction from 7 subprocesses to 4 subprocesses.

In order to obtain the full retrospective splitting bisimulation models of the extension of $\mathrm{ACP}_\epsilon^{\mathrm{cr}}$ with last action conditions, retrospective splitting bisimilarity has to be adapted: in the definition of retrospective splitting bisimulation (see Section 12), the two occurrences of $B(s'_1, {\sim}\alpha', s'_2)$ must be replaced by $B(s'_1, {\sim}\alpha' \sqcap \mathcal{J}_a, s'_2)$.

The operators Π_n^a are reminiscent of the operators Π_n^h. In fact, if we would exclude full retrospective splitting bisimulation models with domain $\mathbb{CTS}_\kappa^{\epsilon\mathrm{r}}/\leftrightarrows^{\mathrm{r}}$ for κ greater than some infinite cardinal λ, Π_n^a could have been replaced by $\Pi_n^{h_a}$, where $h_a \in \mathcal{H}_\lambda^{\mathrm{r}}$ for $a \in \mathsf{A}$ is defined by $h_a(\mathcal{J}_a) = \top$ and $h_a(\mathcal{J}_b) = \bot$ if $a \neq b$ and $h_\delta \in \mathcal{H}_\lambda^{\mathrm{r}}$ is defined by $h_\delta(\mathcal{J}_a) = \bot$.

We conclude with an example of the use of the retrospection operator together with last action conditions.

Example 14.1. The example concerns a service that resembles the services considered in [10, 11]. For any command m from some set M, the service can be requested to process command m and it can be requested to report back what the reply would be to the request to process command m. We suppose that the service can be described by a function $F{:}M^+ \to \{\mathsf{T}, \mathsf{F}, \mathsf{B}\}$ with the property that $F(\alpha) = \mathsf{B} \Rightarrow F(\alpha \frown \langle m \rangle) = \mathsf{B}$. This function is called the *reply* function of the service. Given a reply function F and a command m, the derived reply function of F after processing m, written $\frac{\partial}{\partial m} F$, is defined by $\frac{\partial}{\partial m} F(\alpha) = F(\langle m \rangle \frown \alpha)$. The connection between a reply function F and the service described by it can be understood as follows:

- if $F(\langle m \rangle) \neq \mathsf{B}$, the request to process command m is accepted by the service, the reply is $F(\langle m \rangle)$ and the service proceeds as described by $\frac{\partial}{\partial m} F$;
- if $F(\langle m \rangle) = \mathsf{B}$, the request to process command m is not accepted by the service, the reply is $F(\langle m \rangle)$ and the service proceeds as described by F;
- the request to report back what the reply would be to the request to process command m is always accepted by the service, the reply is $F(\langle m \rangle)$ and the service proceeds as described by F.

Hence, the service can be viewed as the process defined by the guarded recursive specification that consists of an equation

$$P_G = \sum_{m \in M} (r(m) + r(?m)) \cdot s(G(\langle m \rangle)) \cdot (P_{\frac{\partial}{\partial m} G} \trianglelefteq {\sim}\mathcal{J}_{r(m)} \sqcap {-}\mathcal{J}_{s(\mathsf{B})} \trianglerighteq P_G)$$

for each reply function G. Here, we write $r(m)$ for the action of receiving a request to process command m, $r(?m)$ for the action of receiving a request to report back what the reply would be to the request to process command m, and $s(v)$ for the action of sending reply v.

15 Concluding Remarks

We have added the empty process constant to the different extensions of ACP with conditional expressions presented in [9]. In the past, the addition of the empty process constant to ACP was rather problematic. Its current addition to the different extensions of ACP with conditional expressions presented in [9] turns out to present no additional complications.

The addition of the empty process constant to different extensions of ACP in this paper is based on the treatment of the empty process constant in the setting of ACP that is chosen in [5]. If it was based on the treatment of the empty process constant chosen in [19] instead, the addition of the empty process constant to different extensions of ACP in this paper would have been slightly different. For example, with the treatment from [5], no special additional axioms concerning conditional expressions are needed when adding the empty process constant, whereas with the treatment from [19], the special additional axiom $\epsilon \between (\phi :\rightarrow \epsilon) = \phi :\rightarrow \epsilon$ is needed.

In [11], we showed that threads, as found in programming languages such as Java and C#, and services used by them can be viewed as processes that are definable over $\mathrm{ACP}^{\mathrm{c}}$, and that thread-service composition on those processes can be expressed in terms of operators of $\mathrm{ACP}^{\mathrm{c}}$ extended with action renaming. In fact, the termination behaviour of the composition of a thread with the services used by it can be dealt with more directly, and without action renaming, in $\mathrm{ACP}^{\mathrm{c}}_{\epsilon}$.

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 and J. A. Bergstra. Deadlock behaviour in split and ST bisimulation. In I. Castellani and C. Palamidessi, editors, *EXPRESS'98*, volume 16 of *Electronic Notes in Theoretical Computer Science*, pages 101–114. Elsevier, 1998.
5. J. C. M. Baeten and R. J. van Glabbeek. Merge and termination in process algebra. In K. V. Nori, editor, *Proceedings 7th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 287 of *Lecture Notes in Computer Science*, pages 153–172. Springer-Verlag, 1987.
6. J. C. M. Baeten and W. P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1990.
7. J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1/3):109–137, 1984.
8. J. A. Bergstra and C. A. Middelburg. Process algebra for hybrid systems. To appear in *Theoretical Computer Science*, 2005. Preliminary version: Logic Group Preprint Series 225, Department of Philosophy, Utrecht University.

9. 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.
10. J. A. Bergstra and C. A. Middelburg. A thread algebra with multi-level strategic interleaving. In B. Löwe S. B. Cooper and L. Torenvliet, editors, *CiE 2005*, volume 3526 of *Lecture Notes in Computer Science*, pages 35–48. Springer-Verlag, 2005.
11. J. A. Bergstra and C. A. Middelburg. Thread algebra with multi-level strategies. Computer Science Report 05-08, Department of Mathematics and Computer Science, Eindhoven University of Technology, March 2005.
12. J. A. Bergstra, C. A. Middelburg, and Y. S. Usenko. Discrete time process algebra and the semantics of SDL. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebra*, pages 1209–1268. Elsevier, Amsterdam, 2001.
13. C. C. Chang and H. J. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, Amsterdam, third edition, 1990.
14. J. F. Groote and A. Ponse. Proof theory for μCRL: A language for processes with data. In D. J. Andrews, J. F. Groote, and C. A. Middelburg, editors, *Semantics of Specification Languages*, Workshops in Computing Series, pages 232–251. Springer-Verlag, 1994.
15. P. R. Halmos. *Lectures on Boolean Algebras*. Mathematical Studies. Van Nostrand, Princeton, NJ, 1963.
16. C. P. J. Koymans and J. L. M. Vrancken. Extending process algebra with the empty process ϵ. Logic Group Preprint Series 1, Department of Philosophy, Utrecht University, Utrecht, 1985.
17. J. D. Monk and R. Bonnet, editors. *Handbook of Boolean Algebras*, volume 1. Elsevier, Amsterdam, 1989.
18. J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley Series in Logic. Addison-Wesley, Reading, MA, 1967.
19. J. L. M. Vrancken. The algebra of communicating processes with empty process. *Theoretical Computer Science*, 177(2):287–328, 1997.