Process Algebra for Hybrid Systems

J. A. Bergstra [a,b], C. A. Middelburg [c,*]

[a] *Programming Research Group, University of Amsterdam, P.O. Box 41882, 1009 DB Amsterdam, Netherlands*

[b] *Department of Philosophy, Utrecht University, P.O. Box 80126, 3508 TC Utrecht, Netherlands*

[c] *Computing Science Department, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands*

Abstract

We propose a process algebra obtained by extending a combination of the process algebra with continuous relative timing from Baeten and Middelburg [Process Algebra with Timing, Springer, 2002, Chapter 4] and the process algebra with propositional signals from Baeten and Bergstra [Theoretical Computer Science 177 (1977) 381–405]. The proposed process algebra makes it possible to deal with the behaviour of hybrid systems, i.e. systems in which the instantaneous state transitions caused by performing actions are alternated with continuous state evolutions. This process algebra has, in addition to equational axioms, rules to derive equations with the help of real analysis.

Key words: process algebra, hybrid systems, continuous relative timing, propositional signals, state evolutions, state transitions

1 Introduction

There is a rapid growth of interest in systems that exhibit both discrete and continuous behaviour. Such systems, called hybrid systems, are found in many areas, from avionics to consumer electronics. Simple hybrid systems typically consist of a controlling subsystem made up of digital components and a controlled subsystem made up of analog components. The controlling subsystem

* Corresponding author.

Email addresses: janb@phil.uu.nl (J. A. Bergstra), keesm@win.tue.nl (C. A. Middelburg).

URL: http://www.win.tue.nl/~keesm (C. A. Middelburg).

exhibits discrete behaviour and the controlled subsystem exhibits continuous behaviour. In general, the controlling subsystem is embedded in the controlled subsystem without being accessible from the outside. Moreover, the behaviour of the controlling subsystem generally depends on the behaviour of the controlled subsystem and cannot be considered in isolation. More complicated hybrid systems arise, for example, if the controlled subsystem is a distributed system and, for that reason, the controlling subsystem is composed of several distributed controllers and possibly a coordinating supervisor.

It was proposed almost at the outset of the interest for hybrid systems in computer science to model them as hybrid automata [1–3]. Hybrid automata are automata equipped with variables that evolve continuously with time. They can be viewed as a generalization of timed automata [4,5]. The study of hybrid systems in computer science is up to now largely focussed on hybrid automata, in particular on model checking for hybrid automata, i.e. automatic ways for verifying whether a hybrid automaton satisfies a property expected from the hybrid system modelled by it (see e.g. Refs. [6–9]). To the best of our knowledge, little attention is paid to equivalence checking for hybrid automata, i.e. automatic ways for verifying whether two hybrid automata are equivalent in some well-defined sense. Satisfaction of properties expressed in an expressive temporal logic can be automatically verified for a restricted subclass of hybrid automata, known as linear hybrid automata. Conservative approximations are needed for other hybrid automata to make automatic verification possible.

We complement the framework of hybrid automata with a process algebra for hybrid systems. This process algebra, being essentially a calculus of hybrid systems, allows for description and syntax-based analysis of hybrid systems in a compositional way. It comprises:

- mathematical expressions for hybrid systems;
- equational axioms for equational reasoning about hybrid systems;
- rules for lifting results from real analysis to equations about hybrid systems;
- a structural operational semantics of the expressions.

The expressions are constructed by means of operators, each of which corresponds to a distinct and natural way in which hybrid systems can be combined or adapted. The axioms and lifting rules make fully precise how to establish whether two expressions constructed in different ways represent the same hybrid system. The axioms can amongst other things be used to transform an expression into one that is suggestive of a symbolic counterpart of a hybrid automaton. The structural operational semantics induces a transition system for each expression. The transition systems concerned are similar to the ones used for model checking in the setting of hybrid automata. Consequently, those model checking techniques can easily be adapted to the process algebraic setting for hybrid systems.

The process algebra for hybrid systems is also meant to be an algebraic theory which formalizes an important part of our general understanding of hybrid systems. Although the axioms and lifting rules of the process algebra are supported by a model, based on the structural operational semantics, our general understanding of hybrid systems provided the primary justification of the axioms and lifting rules of the process algebra. The process algebra for hybrid systems turns out to be far from a compact theory. The complexity inherent in hybrid systems is also found in the rather large number of axioms. Only a few axioms can be removed because of their derivability from the other axioms. The remaining axioms formalize distinct and basic general properties. In order to condense the theory, the collection of operators has to be restricted. However, this would compromise its relevance to hybrid systems.

Like the framework of hybrid automata, the process algebra for hybrid systems proposed in this paper adopts the view that a hybrid system is a system in which an instantaneous state transition takes place on the system performing an action and a continuous state evolution takes place on the system idling between performing successive actions.

The process algebra for hybrid systems is obtained by extending a combination of two existing extensions of ACP [10], namely the process algebra with continuous relative timing from Ref. [11] and the process algebra with propositional signals from Ref. [12]. A process may idle for some period of time before it performs its next action (instantaneously), in which case the next action is performed after a delay. The process algebra with continuous relative timing covers this aspect of process behaviour. The state of the process may further change continuously during the delay. This is not covered, because the state of processes is kept invisible. In the process algebra with propositional signals, a process can have its state to some extent visible. The basic idea is that the visible part of the state of a process, called the signal emitted by the process, is a proposition. Only discrete state changes, caused by performing actions, are covered.

We introduce a new operator which makes it possible to deal with continuous state changes during delays as well. With the new operator, we can have signals at all points of time during a delay instead of only at its begin and end. For this operator, we have to add some structure to the atomic propositions from which the propositional signals concerned are generated: algebraic and differential equations and inequalities concerning named state components are taken as atomic propositions. We also introduce a new operator which makes it possible to deal better with instantaneous state changes where the state immediately after the change depends upon the state immediately before the change. The resulting process algebra has, in addition to equational axioms, some rules to derive further equations with the help of real analysis. These lifting rules constitute a smooth interface to disciplines such as control engineering where

real analysis is the standard tool. They permit to cast the effects of continuous state changes into equations about processes.

As mentioned before, up to now the study of hybrid systems is largely focussed on hybrid automata. The process algebra proposed in this paper can be regarded as originating from the formalism of hybrid automata in the sense that it has been strongly influenced by the formalism of hybrid automata. This is among other things apparent from the fact that hybrid automata can be faithfully represented using the proposed process algebra in a uniform and direct way. The representation of hybrid automata will be briefly outlined in Section 6. The operational semantics of the proposed process algebra has further been influenced by the concept of abstract phase transition systems from Ref. [13].

Other related work includes the following. A variant of timed CSP [14] in which one can deal with continuous behaviour in a limited way is introduced in Ref. [15]. A variant of the π-calculus [16] in which one can deal with continuous behaviour in another limited way is introduced in Ref. [17]. Those variants of timed CSP and the π-calculus are called hybrid CSP and the ϕ-calculus, respectively. Very shortly after the report version of this paper [18] appeared, another report about an extension of ACP for hybrid systems [19] appeared. That version, called HyPA, does not extend a version of ACP with timing. Thereby, in comparison with the process algebra proposed in this paper, it has some limitations with regard to the description and analysis of hybrid systems. Hybrid CSP, the ϕ-calculus and HyPA will be further discussed in Section 6. Here, we only mention that to the best of our knowledge the first process algebra for hybrid systems is hybrid CSP. There is also work on the description and analysis of hybrid systems in which operations corresponding to ways in which hybrid systems can be combined or adapted are introduced, but which has not yet resulted in an algebraic framework. Notable examples are the work on Charon [20], Masaccio [21], and the HIOA framework [22].

In Ref. [11], a coherent collection of four process algebras with timing, each dealing with timing in a different way, is presented. The time scale on which the time is measured is either discrete or continuous, and the timing of actions is either relative or absolute. There is no other reason to choose for relative timing in this paper but the fact that it is generally considered to be simpler than absolute timing. Various constants and operators of the process algebra with continuous relative timing have counterparts in the other versions from the above-mentioned collection. A notational distinction is made between a constant or operator of one version and its counterparts in another version, by means of different decorations of a common symbol, if they should not be identified in case versions are integrated. So long as one uses a single version, one can safely omit those decorations. However, we refrain from omitting them in this paper because we think that change of notation in a series of technical

publications is undesirable.

We distinguish between a basic process algebra for hybrid systems, which does not cover parallelism and communication, an algebra of communicating processes for hybrid systems, which covers parallelism and communication, and several extensions which are useful or needed in many applications. Two extensions are presented as extensions of the basic process algebra and another one as extension of the algebra of communicating processes. This is only for pedagogical reasons. Integration, which provides for alternative composition over a continuum of differently timed alternatives, and guarded recursion, which allows for the description of (potentially) non-terminating processes, are needed in many applications of the proposed process algebra for hybrid systems. Both integration and guarded recursion are treated as extensions. Localization, which makes it possible to keep discontinuities of named state components local, is useful in various applications. Localization is treated as extension as well.

The structure of this paper is as follows. First of all, we introduce the basic process algebra for hybrid systems (Section 2). Next, we consider the addition of integration and recursion (Section 3). After that, we consider the addition of parallel composition and encapsulation (Section 4). Then, we consider the addition of localization (Section 5). Finally, some concluding remarks are made (Section 6). The application of the process algebra for hybrid systems is regularly illustrated by means of examples.

In the remainder of this paper, we will mostly refer to process algebras by name. The process algebra with continuous relative timing from Ref. [11] and the process algebra with propositional signals from Ref. [12] are known as $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$, respectively. The new process algebra proposed in this paper is called $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$. All of these process algebras are extensions of ACP [23,10]. We will also refer to BPA and BPA_δ, which are names of subtheories of ACP that do not cover parallelism and communication. The difference between them is that BPA does not cover deadlock and BPA_δ does.

The new process algebra proposed in this paper extends $\mathrm{ACP}^{\mathrm{srt}}$. This process algebra was first introduced in Ref. [24]. The motivation of choices made in the design of $\mathrm{ACP}^{\mathrm{srt}}$, as well as a brief comparison with other process algebras with timing, can be found in Ref. [24]. In this paper, we mostly refer to Ref. [11] because, in many respects, it contains a more extensive treatment of $\mathrm{ACP}^{\mathrm{srt}}$. Additional insight in the choices made in the design of $\mathrm{ACP}^{\mathrm{srt}}$ can be gained from Refs. [25,26].

Some familiarity with real analysis is required. The desirable background can, for example, be found in Ref. [27].

2 Basic Process Algebra

In this section, we introduce $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, which is, roughly speaking, the subtheory of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ that does not cover parallelism and communication. Beforehand, we give already an idea of its application by means of an example concerning a water-level monitor. First of all, we introduce $\mathrm{BPA}_{\perp}^{\mathrm{srt}}$, an extension of (a restricted version of) $\mathrm{BPA}^{\mathrm{srt}}$ from Ref. [11] with non-existence like in $\mathrm{BPA}_{\perp}$ from Ref. [12]. Next, we introduce $\mathrm{BPA}_{\mathrm{ps}}^{\mathrm{srt}}$, an extension of $\mathrm{BPA}_{\perp}^{\mathrm{srt}}$ with propositional signals and conditions like in $\mathrm{BPA}_{\mathrm{ps}}$ from Ref. [12]. Finally, we introduce $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, an extension of $\mathrm{BPA}_{\mathrm{ps}}^{\mathrm{srt}}$ with a signal evolution operator and a signal transition operator.

2.1 Example: Water-level Monitor

This section is a sample of the application of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$. It is meant to give a first impression of how one describes the behaviour of hybrid systems in $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$. We describe the behaviour of a water-level monitor. This example is adapted from Ref. [2]. We take the following informal description of the behaviour of the water-level monitor as the starting point of our formal description.

The water-level monitor continuously senses the water level l in a tank and turns a pump on and off, in order to keep it between 0.075 m and 0.300 m. Initially, the water level is 0.075 m and the pump is on. While the pump is on, the water level rises by 0.025 m/s. When the water level becomes 0.250 m, the monitor turns the pump off. While the pump is off, the water level falls by 0.050 m/s. When the water level becomes 0.175 m, the monitor turns the pump on. Naturally, the water level does not change instantaneously when the monitor turns the pump on or off. The change of the status of the pump becomes effective only 2 s later. That is, the pump starts working 2 s after it has been turned on and the pump stops working 2 s after it has been turned off.

The water-level monitor can be formally described using $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ by the fol-

lowing equations:

$$
\begin{aligned}
W &= (l = 0.075) \trianglerighteq W^{\mathrm{on}} \,, \\
W^{\mathrm{on}} &= (l \leq 0.250 \wedge \dot{l} = 0.025) \trianglerighteq \\
&\qquad \sigma^*_{\mathsf{rel}}\Big((l = 0.250) \mathbin{:\!\rightarrow} \Big((l^\bullet = {}^\bullet l) \trianglerighteq \widetilde{\widetilde{\textit{turn-off}}} \cdot W^{\mathrm{on}'}\Big)\Big) \,, \\
W^{\mathrm{on}'} &= (l \leq 0.300 \wedge \dot{l} = 0.025) \trianglerighteq \sigma^2_{\mathsf{rel}}\Big((l^\bullet = {}^\bullet l) \trianglerighteq \widetilde{\widetilde{\textit{stop}}} \cdot W^{\mathrm{off}}\Big) \,, \\
W^{\mathrm{off}} &= (l \geq 0.175 \wedge \dot{l} = -0.050) \trianglerighteq \\
&\qquad \sigma^*_{\mathsf{rel}}\Big((l = 0.175) \mathbin{:\!\rightarrow} \Big((l^\bullet = {}^\bullet l) \trianglerighteq \widetilde{\widetilde{\textit{turn-on}}} \cdot W^{\mathrm{off}'}\Big)\Big) \,, \\
W^{\mathrm{off}'} &= (l \geq 0.075 \wedge \dot{l} = -0.050) \trianglerighteq \sigma^2_{\mathsf{rel}}\Big((l^\bullet = {}^\bullet l) \trianglerighteq \widetilde{\widetilde{\textit{start}}} \cdot W^{\mathrm{on}}\Big) \,.
\end{aligned}
$$

At this stage, we cannot further explain this description. However, note that it appears to be a fairly direct representation of the informal description given above ($\dot{l}$ stands for the derivative of l). In addition to constants and operators of $\mathrm{BPA}^{\mathrm{srt}}$ [11] and $\mathrm{BPA}_{\mathrm{ps}}$ [12], the signal transition operator $\trianglerighteq$ and the signal evolution operator $\trianglerighteq$ are used. These new operators are needed to make precise that the water level does not change instantaneously at the points of time at which the monitor turns the pump on or off or the pump starts or stops working and that the water level changes continuously as described above during the periods in between.

2.2 $\mathrm{BPA}^{\mathrm{srt}}$ *with Non-existence*

The atomic processes are undelayable actions. Let a be an action. Then *undelayable action* a, written $\tilde{\tilde{a}}$, is the process that immediately performs action a at the current point of time and then terminates successfully. Actions are idealized in the sense that they are treated as if they are performed instantaneously.

The basic way of timing processes is relative delay. Let P be a process and $r \in \mathbb{R}^{\geq}$. Then the *relative delay* of P for a period of time r, written $\sigma^r_{\mathsf{rel}}(P)$, is the process that idles for a period of time r and then behaves like P. In other words, it is P after a delay of r time units.

The basic ways of combining processes are alternative composition and sequential composition. Let P_1 and P_2 be processes. Then the *alternative composition* of P_1 and P_2, written $P_1 + P_2$, is the process that behaves either like P_1 or like P_2, but not both. In other words, there is an arbitrary choice between P_1 and P_2. The choice is resolved on one of them performing its first action, and not otherwise. Consequently, the choice between two idling processes will always

be postponed until at least one of the processes can perform its first action. Only when both processes cannot idle any longer, further postponement is not an option. If the choice has not yet been resolved when one of the processes cannot idle any longer, the choice will simply not be resolved in its favour. The *sequential composition* of P_1 and P_2, written $P_1 \cdot P_2$, is the process that first behaves like P_1, but when P_1 terminates successfully it continues by behaving like P_2. That is, P_1 is followed by P_2. If P_1 never terminates successfully, the sequential composition of P_1 and P_2 will behave like P_1.

In order to deal with unsuccessful termination, we need an additional process that is neither capable of performing any action nor capable of idling beyond the current point of time. This process, written $\tilde{\tilde{\delta}}$, is called *undelayable deadlock*.

We further introduce a process that is considered to be in an inconsistent state from its start. We need this process further on when we introduce propositional signals (it corresponds to a process that emits a signal that cannot hold). It is common to consider a process with such an inconsistency to be non-existent. Therefore, this process, written $\bot$, is (rather contradictory) called the *non-existent* process. Like undelayable deadlock, $\bot$ is neither capable of performing any action nor capable of idling beyond the current point of time. Moreover, a choice involving the non-existent process and the non-existent process followed by another process are non-existent as well.

For convenience later on, we also add an auxiliary operator: ν_{rel}. The operator ν_{rel} is interpreted as relative undelayable time-out. Let P be a process. The *relative undelayable time-out* of P, written $\nu_{\mathsf{rel}}(P)$, behaves like the part of P that starts to perform actions at the current point of time if P is capable of performing actions at the current point of time. Otherwise, it behaves like undelayable deadlock. That is, the relative undelayable time-out keeps P entirely from idling.

The process algebra introduced here features urgent actions. This means that it is possible for two or more actions to be performed consecutively at the same point of time. In Ref. [25], it is shown, using the finite elements of the nonstandard extension of $\mathbb{R}^{\geq}$ as time domain, that actions that are performed consecutively at the same point of time in $\mathbb{R}^{\geq}$, say p, can be considered to be performed at different points in time that are infinitely close to p. Other process algebras featuring urgent actions include the ACP-style process algebras with timing presented in Ref. [11], ATP [28], the different versions of CCS with timing [29–31], Timed CSP [14], TIC [32], and TPL [33].

We shall henceforth use $x, y, x', y', \ldots$ as variables ranging over processes. Furthermore, we shall henceforth use $p, q, r, \ldots$ to stand for arbitrary closed terms denoting non-negative real numbers, and $a, b, c, \ldots$ to stand for arbitrary ac-

Table 1
Axioms of $\mathrm{BPA}_{\bot}^{\mathrm{srt}}$ ($a \in \mathsf{A}_\delta$, $p, q \geq 0$, $r > 0$)

$x + y = y + x$	A1	$\sigma_{\mathsf{rel}}^0(x) = x$	SRT1
$(x + y) + z = x + (y + z)$	A2	$\sigma_{\mathsf{rel}}^p(\sigma_{\mathsf{rel}}^q(x)) = \sigma_{\mathsf{rel}}^{p+q}(x)$	SRT2
$x + x = x$	A3	$\sigma_{\mathsf{rel}}^p(x) + \sigma_{\mathsf{rel}}^p(y) = \sigma_{\mathsf{rel}}^p(x + y)$	SRT3
$(x + y) \cdot z = (x \cdot z) + (y \cdot z)$	A4	$\sigma_{\mathsf{rel}}^p(x) \cdot y = \sigma_{\mathsf{rel}}^p(x \cdot y)$	SRT4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5		
$x + \tilde{\tilde{\delta}} = x$	A6SR	$\nu_{\mathsf{rel}}(\tilde{a}) = \tilde{a}$	SRU1
$\tilde{\tilde{\delta}} \cdot x = \tilde{\tilde{\delta}}$	A7SR	$\nu_{\mathsf{rel}}(\sigma_{\mathsf{rel}}^r(x)) = \tilde{\tilde{\delta}}$	SRU2
		$\nu_{\mathsf{rel}}(x + y) = \nu_{\mathsf{rel}}(x) + \nu_{\mathsf{rel}}(y)$	SRU3
$x + \bot = \bot$	NE1	$\nu_{\mathsf{rel}}(x \cdot y) = \nu_{\mathsf{rel}}(x) \cdot y$	SRU4
$\bot \cdot x = \bot$	NE2		
$\tilde{a} \cdot \bot = \tilde{\tilde{\delta}}$	NE3SR	$\nu_{\mathsf{rel}}(\bot) = \bot$	NESRU

tions.

It is assumed that a fixed but arbitrary set A of *actions* has been given. We write A_δ for $\mathsf{A} \cup \{\delta\}$. An important convention is that we use $a, b, c, \ldots$ to stand for elements of A_δ in the context of equations and for elements of A in the context of transition rules (used for describing structural operational semantics), unless explicitly indicated otherwise.

The axioms of $\mathrm{BPA}_{\bot}^{\mathrm{srt}}$ are the equations given in Table 1. Many axioms in this table and coming ones are actually axiom schemas. In this table, for example, a stands for an arbitrary action, and p and q stand for arbitrary closed terms denoting non-negative real numbers. Axioms A1–A5 are the axioms of BPA. Axioms A6SR and A7SR are simple reformulations of axioms A6 and A7 of BPA_δ. The constant δ has been replaced by the constant $\tilde{\tilde{\delta}}$. For a detailed introduction to BPA and BPA_δ, see Ref. [10]. Axioms SRT1 and SRT2 point out that a delay of 0 time units has no effect and that consecutive delays count up. Axiom SRT3, called the time-factorization axiom, shows that a delay by itself cannot determine a choice. Axiom SRT4 reflects that timing is relative. Axioms SRU1–SRU4 make clear that relative undelayable time-out prevents a process from idling at the start. Axioms NE1 and NE2 express that a choice involving the non-existent process and the non-existent process followed by another process are non-existent as well. Axiom NE3SR expresses that going on as $\bot$ after performing an action is impossible. Axiom NESRU expresses that keeping $\bot$ from idling has no effect.

Note that the following interesting equations are derivable ($p \geq 0$, $r > 0$):

$$\sigma_{\mathsf{rel}}^{p+r}(x) + \sigma_{\mathsf{rel}}^r(\tilde{\tilde{\delta}}) = \sigma_{\mathsf{rel}}^{p+r}(x) \;,$$
$$\sigma_{\mathsf{rel}}^{p+r}(x) + \sigma_{\mathsf{rel}}^r(\bot) = \sigma_{\mathsf{rel}}^r(\bot) \;.$$

The axioms of $\mathrm{BPA}^{\mathrm{srt}}_{\perp}$ are essentially the axioms of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\perp}$ with on top of that axiom NESRU concerning the effect of relative undelayable time-out on the non-existent process. Axiom NESRU is the only additional axiom. In particular, we do not have any additional axiom concerning the effect of relative delay on the non-existent process. The process $\sigma^r_{\mathsf{rel}}(\perp)$ $(r > 0)$ is considered to be capable of idling, but only till arbitrarily close to the point of time that is reached after a period of time r. Thus, just like after performing an action, it is impossible to go on as $\perp$ after idling for a period of time. However, there are no additional identifications of processes possible as a result of the interaction between relative delay and the non-existent process.

Throughout this paper, the need to use parentheses is reduced by using the associativity of the operators $+$ and $\cdot$, and by ranking the precedence of the binary operators. We adhere to the following precedence rules: (i) the operator $+$ has lower precedence than all others, (ii) the operator $\cdot$ has higher precedence than all others, and (iii) all other operators have the same precedence. Moreover, we shall use the notation $\sum_{i\in\mathcal{I}} t_i$, where $\mathcal{I} = \{i_1, \ldots, i_n\}$ and $t_{i_1}, \ldots, t_{i_n}$ are terms denoting processes, for $t_{i_1} + \ldots + t_{i_n}$. The convention is that $\sum_{i\in\mathcal{I}} t_i$ stands for $\tilde{\delta}$ if $\mathcal{I} = \emptyset$.

2.3 $\mathrm{BPA}^{\mathrm{srt}}$ *with Propositional Signals*

Propositions are used both as signals that are emitted by processes and as conditions that are imposed on processes to proceed. Condition testing is looked upon as signal inspection. The intuition is that the signal emitted by a process, as well as each of its logical consequences, holds at the start of the process. The signal emitted by a process is also called the signal of the process.

The basic ways of dealing with propositions are signal emission and conditional proceeding. Let P be a process and ψ be a proposition. Then P *emitting signal* ψ, written $\psi \mathbin{\blacktriangleright} P$, is the process that behaves like P and moreover emits signal ψ; and P *proceeding conditionally* on ψ, written $\psi :\!\rightarrow P$, is the process that behaves like P if proposition ψ holds at its start, and otherwise behaves like undelayable deadlock.

It is assumed that a fixed but arbitrary set P_{at} of *atomic propositions* has been given. Propositions over P_{at} are constructed in the usual way with constants T, F and the various logical connectives ($\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$).

We shall henceforth use $\psi, \psi', \ldots$ to stand for arbitrary (state) propositions over P_{at}.

In derivations we may always use logical equivalences of propositional logic. So we are actually using equivalence classes of propositions, with respect to

Table 2
Additional axioms for $\mathrm{BPA}_{\mathrm{ps}}^{\mathrm{srt}}$

$\mathsf{T} :\!\rightarrow x = x$	GC1
$\mathsf{F} :\!\rightarrow x = \tilde{\tilde{\delta}}$	GC2SR
$\psi :\!\rightarrow \tilde{\tilde{\delta}} = \tilde{\tilde{\delta}}$	GC3SR
$\psi :\!\rightarrow (x + y) = \psi :\!\rightarrow x + \psi :\!\rightarrow y$	GC4
$\psi :\!\rightarrow x \cdot y = (\psi :\!\rightarrow x) \cdot y$	GC5
$\psi :\!\rightarrow (\psi' :\!\rightarrow x) = (\psi \wedge \psi') :\!\rightarrow x$	GC6
$(\psi \vee \psi') :\!\rightarrow x = \psi :\!\rightarrow x + \psi' :\!\rightarrow x$	GC7
$\mathsf{T} \mathbin{\triangle} x = x$	SE1
$\mathsf{F} \mathbin{\triangle} x = \bot$	SE2
$\psi \mathbin{\triangle} x + y = \psi \mathbin{\triangle} (x + y)$	SE3
$(\psi \mathbin{\triangle} x) \cdot y = \psi \mathbin{\triangle} x \cdot y$	SE4
$\psi \mathbin{\triangle} (\psi' \mathbin{\triangle} x) = (\psi \wedge \psi') \mathbin{\triangle} x$	SE5
$\psi \mathbin{\triangle} (\psi :\!\rightarrow x) = \psi \mathbin{\triangle} x$	SE6
$\psi :\!\rightarrow (\psi' \mathbin{\triangle} x) = (\psi \rightarrow \psi') \mathbin{\triangle} (\psi :\!\rightarrow x)$	SE7
$\nu_{\mathsf{rel}}(\psi :\!\rightarrow x) = \psi :\!\rightarrow \nu_{\mathsf{rel}}(x)$	PSSRU1
$\nu_{\mathsf{rel}}(\psi \mathbin{\triangle} x) = \psi \mathbin{\triangle} \nu_{\mathsf{rel}}(x)$	PSSRU2

logical equivalence, instead of the propositions themselves.

The axioms of $\mathrm{BPA}_{\mathrm{ps}}^{\mathrm{srt}}$ are the equations given in Tables 1 and 2. Axioms GC1–GC7 and SE1–SE7 are simple reformulations of corresponding axioms of $\mathrm{BPA}_{\mathrm{ps}}$ (see Ref. [12]). The constant δ has again been replaced by the constant $\tilde{\tilde{\delta}}$. Axiom SE2 expresses that a process emitting the signal F is non-existent. Axioms SE6 and SE7 represent the interaction between signal emission and conditional proceeding. Axiom SE6 reflects that condition testing is looked upon as signal inspection. Axiom SE7 points out that if a proposition holds at the start of a process and that process is proceeding conditional on another proposition then at the start of the whole the former proposition holds or the latter proposition does not hold. Axioms PSSRU1 and PSSRU2 are new axioms concerning the interaction of relative undelayable time-out with conditional proceeding and signal emission.

Note that axioms NE1, NE2 and NESRU are derivable from axioms A1, SE2, SE3, SE4 and PSSRU2. Note further that the following generalizations of

axioms SE3 and SE6 are derivable:

$$\psi \mathbin{\leftthreetimes} x + \psi' \mathbin{\leftthreetimes} y = (\psi \wedge \psi') \mathbin{\leftthreetimes} (x + y) ,$$
$$(\psi \wedge \psi') \mathbin{\leftthreetimes} (\psi :\!\rightarrow x) = (\psi \wedge \psi') \mathbin{\leftthreetimes} x ,$$
$$\psi \mathbin{\leftthreetimes} ((\psi \wedge \psi') :\!\rightarrow x) = \psi \mathbin{\leftthreetimes} (\psi' :\!\rightarrow x) .$$

Note also that the following interesting specialization of axiom SE3 is derivable:

$$\psi \mathbin{\leftthreetimes} \tilde{\tilde{\delta}} + x = \psi \mathbin{\leftthreetimes} x .$$

Useful derivable equations concerning the non-existing process are:

$$\psi \mathbin{\leftthreetimes} \bot = \bot ,$$
$$\psi :\!\rightarrow \bot = \neg\psi \mathbin{\leftthreetimes} \tilde{\tilde{\delta}} .$$

The axioms of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{ps}}$ are essentially the axioms of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\mathrm{ps}}$ with on top of that axiom NESRU concerning the effect of relative undelayable time-out on the non-existent process and axioms PSSRU1 and PSSRU2 concerning the interaction of relative undelayable time-out with conditional proceeding and signal emission. Axioms NESRU, PSSRU1 and PSSRU2 are the only additional axioms. In particular, we do not have any additional axiom concerning the interaction of relative delay with conditional proceeding and signal emission. Conditional proceeding is non-waiting. Therefore, we do not have $\psi :\!\rightarrow \sigma^r_{\mathsf{rel}}(x) = \sigma^r_{\mathsf{rel}}(\psi :\!\rightarrow x)$. Signal emission is non-persistent, both over performing an action and idling for a period of time. Therefore, we do not have $\psi \mathbin{\leftthreetimes} \sigma^r_{\mathsf{rel}}(x) = \sigma^r_{\mathsf{rel}}(\psi \mathbin{\leftthreetimes} x)$.

In Ref. [34], a counterpart of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{ps}}$ with discrete relative timing is presented, which includes a non-waiting version of the conditional proceeding operator as well as a waiting version. In that paper, the symbol $:\!\rightarrow$ is used for the waiting version. The reason for this was that in a natural embedding of $\mathrm{BPA}_{\mathrm{ps}}$, the conditional proceeding operator of $\mathrm{BPA}_{\mathrm{ps}}$, for which the symbol $:\!\rightarrow$ is used as well, corresponds to the waiting version. In the current paper, in which no waiting version is introduced, the symbol $:\!\rightarrow$ is used for the non-waiting version. This is done because the axioms concerning the non-waiting version are essentially the same as the axioms concerning the conditional proceeding operator of $\mathrm{BPA}_{\mathrm{ps}}$.

In Sections 2.2 and 2.3, existing (basic) process algebras were simply joined. No new constants or operators were added. With $\mathrm{BPA}^{\mathrm{srt}}$ for hybrid systems, it becomes more interesting because new operators, which make it possible to deal with the behaviour of hybrid systems, are introduced.

In the case of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$, we add some structure to the atomic propositions of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{ps}}$. That is, algebraic and differential equations and inequalities concerning named state components, called state variables, are taken as atomic propositions. From now on, we will call them atomic state propositions. In conformity with that, the propositions that can be constructed from atomic state propositions will be called state propositions. They will be defined precisely later on. State variables are real-valued functions of time. Their values may change both instantaneously at the points of time at which an action is performed and continuously during the periods in between.

In order to deal with continuous state evolutions, the signal evolution operator is introduced. Let P be a process, V be a set of state variables, and ϕ be a state proposition. Then P in *evolution* according to ϕ with V smooth, written $\phi \mathbin{⌜\!\blacktriangledown}_V P$, is the process P of which the emitted signal changes continuously till it performs its first action in such a way that ϕ is satisfied and without discontinuities for the state variables in V. If the first action is performed immediately, signal evolution does not take its signal changing effect. What remains in such cases is that P emits signal ϕ at the start.

In order to deal with instantaneous state transitions, the signal transition operator is introduced. This operator requires transition propositions, i.e. propositions concerning the values of the state variables immediately before and after a transition, instead of state propositions. Transition propositions, just as state propositions, will be defined precisely later on. Let P be a process and χ be a transition proposition. Then P in *transition* according to χ, written $\chi \mathbin{⌜\!\blacktriangledown} P$, is the process P of which the emitted signal changes instantaneously over performing its first action in such a way that χ is satisfied, if it performs its first action immediately. Otherwise, signal transition does not take its signal changing effect. In either case, the process $\chi \mathbin{⌜\!\blacktriangledown} P$ behaves like undelayable deadlock if there is no transition satisfying χ possible at the start of P.

The signal transition operator supersedes the terminal signal emission operator from Ref. [12]. The terminal signal emission operator is in general inadequate if the state immediately after a transition depends upon the state immediately before the transition.

It is assumed that a fixed but arbitrary set V of *state variables* has been given. For each state variable $v \in \mathsf{V}$, we introduce an additional state variable

$\dot{v}$, standing for the derivative of v. We write $\dot{\mathsf{V}}$ for $\{\dot{v} \mid v \in \mathsf{V}\}$. It is further assumed that a set of constants, arithmetic operators and relational operators of real arithmetic, including the basic ones $(0, 1, +, -, \cdot, ^{-1}, <)$, has been given. The set of *state expressions* is inductively defined by the following formation rules:

- each state variable $v \in \mathsf{V} \cup \dot{\mathsf{V}}$ is a state expression;
- each constant c is a state expression;
- if o is an arithmetic operator of arity n and $s_1, \ldots, s_n$ are state expressions, then $o(s_1, \ldots, s_n)$ is a state expression.

The set of *atomic state propositions* is inductively defined by the following formation rules:

- if s_1 and s_2 are state expressions, then $s_1 = s_2$ is an atomic state proposition;
- if π is a relational operator of arity n, and $s_1, \ldots, s_n$ are state expressions, then $\pi(s_1, \ldots, s_n)$ is an atomic state proposition.

In the case of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$, we take the set of atomic state propositions as the set P_{at}. State propositions are not propositions over P_{at} in the customary meaning. There are two additional ways to construct state propositions, which will be described after the introduction of transition propositions.

For each state variable $v \in \mathsf{V} \cup \dot{\mathsf{V}}$, we further introduce two additional state variables ${}^\bullet v$ and $v^\bullet$, standing for the state variable v immediately before and immediately after a transition. We write ${}^\bullet\mathsf{V}$ for $\{{}^\bullet v \mid v \in \mathsf{V} \cup \dot{\mathsf{V}}\}$ and $\mathsf{V}^\bullet$ for $\{v^\bullet \mid v \in \mathsf{V} \cup \dot{\mathsf{V}}\}$. The set of *transition expressions* is inductively defined by the following formation rules:

- each state variable $v \in {}^\bullet\mathsf{V} \cup \mathsf{V}^\bullet$ is a transition expression;
- each constant c is a transition expression;
- if o is an arithmetic operator of arity n and $t_1, \ldots, t_n$ are transition expressions, then $o(t_1, \ldots, t_n)$ is a transition expression.

The set of *atomic transition propositions* is inductively defined by the following formation rules:

- if t_1 and t_2 are transition expressions, then $t_1 = t_2$ is an atomic transition proposition;
- if π is a relational operator of arity n, and $t_1, \ldots, t_n$ are transition expressions, then $\pi(t_1, \ldots, t_n)$ is an atomic transition proposition.

Transition propositions are constructed from atomic transition propositions in the usual way with constants T, F and the various logical connectives, and in addition according to the following formation rule:

- if ψ is a state proposition, then ${}^\bullet\psi$ and $\psi^\bullet$ are transition propositions.

The proposition ${}^\bullet\psi$ is satisfied exactly by the transitions from a state in which ψ holds, and the proposition $\psi^\bullet$ is satisfied exactly by the transitions to a state in which ψ holds.

We are now able to come back to the construction of state propositions. State propositions are constructed from atomic state propositions in the usual way with constants T, F and the various logical connectives, and in addition according to the following formation rule:

- if χ is a transition proposition, then ${}^\circ\chi$ and χ° are state propositions.

The proposition ${}^\circ\chi$ holds exactly in the states from which a transition satisfying χ is possible, and the proposition χ° holds exactly in the states to which a transition satisfying χ is possible.

We write $\mathcal{P}_{\mathsf{st}}$ for the set of all state propositions, and we write $\mathcal{P}_{\mathsf{tr}}$ for the set of all transition propositions.

We adhere to the customary notational conventions for real arithmetic. For example, we shall generally use infix notation for binary operators, and prefix notation for unary operators.

Let ψ be a state proposition. Then we write $\psi[{}^\bullet\mathsf{V}/\mathsf{V}]$ and $\psi[\mathsf{V}^\bullet/\mathsf{V}]$ for ψ with, for each $v \in \mathsf{V} \cup \dot{\mathsf{V}}$, each occurrence of v in ψ replaced by ${}^\bullet v$ and $v^\bullet$, respectively. As to be expected, satisfaction of transition propositions is defined in Section 2.6 such that the transition proposition ${}^\bullet\psi$ is satisfied by the same transitions as $\psi[{}^\bullet\mathsf{V}/\mathsf{V}]$, and the transition proposition $\psi^\bullet$ is satisfied by the same transitions as $\psi[\mathsf{V}^\bullet/\mathsf{V}]$. Moreover, satisfaction of state propositions is defined such that the state propositions ${}^\circ(\psi^\bullet)$ and $({}^\bullet\psi)^\circ$ hold in a state if there exists a state in which ψ holds. In other words, those state propositions express satisfiability of ψ.

Let ϕ and χ be a state proposition and a transition proposition, respectively. Then we write $\mathrm{V}(\phi)$ for the set of all $v \in \mathsf{V}$ with v or $\dot{v}$ occurring in ϕ, and $\mathrm{V}(\chi)$ for the set of all $v \in \mathsf{V}$ with ${}^\bullet v$, ${}^\bullet\dot{v}$, $v^\bullet$ or $\dot{v}^\bullet$ occurring in χ.

We shall use the notation $\phi \upharpoonright t$ for $\phi \upharpoonright_{\mathrm{V}(\phi)} t$. We shall also use the notation $\mathcal{C}_V$, where $V \subseteq \mathsf{V}$, for $\bigwedge_{v \in V}(v^\bullet = {}^\bullet v \wedge \dot{v}^\bullet = {}^\bullet\dot{v})$.

We shall henceforth use $\psi, \psi', \ldots$ as well as $\phi, \phi', \ldots$ to stand for arbitrary state propositions, and $\chi, \chi', \ldots$ to stand for arbitrary transition propositions. In general, we use $\psi, \psi', \ldots$ only in the cases where the state propositions occur solely as an operand of the signal emission operator and/or the conditional proceeding operator. Furthermore, we shall henceforth use $v, v', \ldots$ to stand for

arbitrary elements of V, unless explicitly mentioned otherwise, and $V, V', \ldots$ to stand for arbitrary subsets of V.

In derivations, we may always use equivalences of state propositions and equivalences of transition propositions that are results of real arithmetic. So, like in the case of $\mathrm{BPA}_{\mathrm{ps}}^{\mathrm{srt}}$, we are actually using equivalence classes of propositions instead of the propositions themselves.

The axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ are the equations given in Tables 1, 2, and 3. In Table 3, we use a to stand for elements of A. Axioms HSE1–HSE12 and HST1–HST12 show that signal evolution and signal transition take effect over what takes place first, which is either performing an action or idling for a period of time, and that signal evolution keeps taking effect in the case of idling till the first action is performed. However, equations expressing how signal evolution actually changes the signal of a process during idling cannot be derived using the axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$. The reason for this is that the equations concerned can only be derived with the help of real analysis. We will introduce some rules for this kind of derivations in Section 2.5. Axioms HSE3–HSE6 show that signal evolution only takes its signal changing effect in the case where idling takes place first. Together with axiom HSE9, they also indicate that the state proposition concerned always hold at the start of the process concerned, even in the case where nothing can take place. Axioms HST3–HST6 show that signal transition only takes its signal changing effect in the case where performing an action takes place first. Together with axiom HST9, they also indicate that the process concerned will always behave like undelayable deadlock if there is no transition satisfying the transition proposition concerned possible at its start, even in the case where idling takes place first. Axioms HSE1 and HST1 are reminiscent of axioms SE1 and GC1, respectively; and axioms HSE2 and HST2 are reminiscent of axioms SE2 and GC2, respectively (all closed substitution instances of HSE2 and HST2 are derivable from the other axioms). Axiom HSE13 expresses that in the case of a choice between two idling processes the signals of the idling processes change jointly until one of them performs its first action. It would have been very inconvenient to express this without the relative undelayable time-out operator. Axioms HST13 and HST14 show that there are cases in which signal emission and conditional proceeding can be eliminated in favour of signal transition. Axioms HSSRU1 and HSSRU2 show that signal evolution and signal transition take effect over what takes place first, also in the presence of relative undelayable time-out.

Note that axioms HSE4 and HST6 are derivable specializations of axioms HSE12 and HST12; and that axiom HSE5 is derivable from axioms HSE4 and HSE8. Note further that the following specializations of axioms HSE6 and

Table 3
Additional axioms for $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ ($a \in \mathsf{A}$, $r > 0$)

$\mathsf{T} \mathbin{\upharpoonright}_{\emptyset} x = x$	HSE1
$\mathsf{F} \mathbin{\upharpoonright}_V x = \bot$	HSE2
$\phi \mathbin{\upharpoonright}_V \tilde{\tilde{\delta}} = \phi \mathbin{\curlywedge} \tilde{\tilde{\delta}}$	HSE3
$\phi \mathbin{\upharpoonright}_V \tilde{\tilde{a}} = \phi \mathbin{\curlywedge} \tilde{\tilde{a}}$	HSE4
$\phi \mathbin{\upharpoonright}_V \tilde{\tilde{a}} \cdot x = \phi \mathbin{\curlywedge} \tilde{\tilde{a}} \cdot x$	HSE5
$\phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(x) = \phi \mathbin{\upharpoonright}_V (\phi \mathbin{\curlywedge} \sigma^r_{\mathsf{rel}}(\phi \mathbin{\upharpoonright}_V x))$	HSE6
$\phi \mathbin{\upharpoonright}_V (x + y) = \phi \mathbin{\upharpoonright}_V x + \phi \mathbin{\upharpoonright}_V y$	HSE7
$\phi \mathbin{\upharpoonright}_V x \cdot y = (\phi \mathbin{\upharpoonright}_V x) \cdot y$	HSE8
$\phi \mathbin{\upharpoonright}_V (\psi :\rightarrow x) = \phi \mathbin{\curlywedge} (\psi :\rightarrow (\phi \mathbin{\upharpoonright}_V x))$	HSE9
$\phi \mathbin{\upharpoonright}_V (\psi \mathbin{\curlywedge} x) = \psi \mathbin{\curlywedge} (\phi \mathbin{\upharpoonright}_V x)$	HSE10
$\phi \mathbin{\upharpoonright}_V (\phi' \mathbin{\upharpoonright}_{V'} x) = (\phi \wedge \phi') \mathbin{\upharpoonright}_{V \cup V'} x$	HSE11
$\phi \mathbin{\upharpoonright}_V (\chi \mathbin{\upharpoonright} \tilde{\tilde{a}}) = \phi \mathbin{\curlywedge} (\chi \mathbin{\upharpoonright} \tilde{\tilde{a}})$	HSE12
$\phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(x) + \phi' \mathbin{\upharpoonright}_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(y)) = \phi \mathbin{\upharpoonright}_V (\sigma^r_{\mathsf{rel}}(x) + \phi' \mathbin{\upharpoonright}_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(y)))$	HSE13
$\mathsf{T} \mathbin{\upharpoonright} x = x$	HST1
$\mathsf{F} \mathbin{\upharpoonright} x = \tilde{\tilde{\delta}}$	HST2
$\chi \mathbin{\upharpoonright} \tilde{\tilde{\delta}} = \tilde{\tilde{\delta}}$	HST3
$\chi \mathbin{\upharpoonright} \tilde{\tilde{a}} = \chi \mathbin{\upharpoonright} ({}^\circ\chi :\rightarrow \tilde{\tilde{a}})$	HST4
$\chi \mathbin{\upharpoonright} \tilde{\tilde{a}} \cdot x = \chi \mathbin{\upharpoonright} ({}^\circ\chi :\rightarrow \tilde{\tilde{a}} \cdot (\chi^\circ \mathbin{\curlywedge} x))$	HST5
$\chi \mathbin{\upharpoonright} \sigma^r_{\mathsf{rel}}(x) = {}^\circ\chi :\rightarrow \sigma^r_{\mathsf{rel}}(x)$	HST6
$\chi \mathbin{\upharpoonright} (x + y) = \chi \mathbin{\upharpoonright} x + \chi \mathbin{\upharpoonright} y$	HST7
$\chi \mathbin{\upharpoonright} x \cdot y = (\chi \mathbin{\upharpoonright} x) \cdot y$	HST8
$\chi \mathbin{\upharpoonright} (\psi :\rightarrow x) = \psi :\rightarrow (\chi \mathbin{\upharpoonright} x)$	HST9
$\chi \mathbin{\upharpoonright} (\psi \mathbin{\curlywedge} x) = ({}^\circ\chi \rightarrow \psi) \mathbin{\curlywedge} (\chi \mathbin{\upharpoonright} x)$	HST10
$\chi \mathbin{\upharpoonright} (\chi' \mathbin{\upharpoonright} \tilde{\tilde{a}}) = (\chi \wedge \chi') \mathbin{\upharpoonright} \tilde{\tilde{a}}$	HST11
$\chi \mathbin{\upharpoonright} (\phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(x)) = {}^\circ\chi :\rightarrow (\phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(x))$	HST12
$\psi :\rightarrow \tilde{\tilde{a}} = {}^\bullet\psi \mathbin{\upharpoonright} \tilde{\tilde{a}}$	HST13
$\tilde{\tilde{a}} \cdot (\psi \mathbin{\curlywedge} x) = \psi^\bullet \mathbin{\upharpoonright} \tilde{\tilde{a}} \cdot x$	HST14
$\nu_{\mathsf{rel}}(\phi \mathbin{\upharpoonright}_V x) = \phi \mathbin{\upharpoonright}_V \nu_{\mathsf{rel}}(x)$	HSSRU1
$\nu_{\mathsf{rel}}(\chi \mathbin{\upharpoonright} x) = \chi \mathbin{\upharpoonright} \nu_{\mathsf{rel}}(x)$	HSSRU2

HST5 are derivable ($a \in \mathsf{A}$, $r > 0$):

$$\phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(x) = \phi \mathbin{\upharpoonright}_V (\phi \mathbin{\curlywedge} \sigma^r_{\mathsf{rel}}(x)) ,$$
$$\phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(x) = \phi \mathbin{\upharpoonright}_V \sigma^r_{\mathsf{rel}}(\phi \mathbin{\upharpoonright}_V x) ,$$
$$\chi \mathbin{\upharpoonright} \tilde{\tilde{a}} \cdot x = \chi \mathbin{\upharpoonright} ({}^\circ\chi :\rightarrow \tilde{\tilde{a}} \cdot x) ,$$
$$\chi \mathbin{\upharpoonright} \tilde{\tilde{a}} \cdot x = \chi \mathbin{\upharpoonright} \tilde{\tilde{a}} \cdot (\chi^\circ \mathbin{\curlywedge} x) .$$

Note also that the following specializations of axiom HSE13 are derivable ($r > 0$):

$$\phi \mathbin{\curvearrowright_V} \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}) + \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(x)) = \phi \mathbin{\curvearrowright_V} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(x)) \;,$$
$$\phi \mathbin{\curvearrowright_V} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(x)) + \phi' \mathbin{\curvearrowright_{V'}} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(y))$$
$$= (\phi \wedge \phi') \mathbin{\curvearrowright_{V \cup V'}} (\sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(x)) + \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(y))) \;.$$

The following interesting equations are derivable for all closed terms t ($r > 0$):

$$\phi \mathbin{\curvearrowright_V} \sigma^r_{\mathsf{rel}}(t) = \phi \mathbin{\curvearrowright_V} (\phi \mathbin{\curvearrowright} \sigma^r_{\mathsf{rel}}(\phi \mathbin{\curvearrowright} t)) \;,$$
$$\phi \mathbin{\curvearrowright_V} t = \phi \mathbin{\curvearrowright} (\phi \mathbin{\curvearrowright_V} t) \;,$$
$$\chi \mathbin{\curvearrowright} t = {}^\circ\chi :\to (\chi \mathbin{\curvearrowright} t) \;.$$

The following derivable equation shows how signal transition changes the signal of a process over performing an action:

$$\psi \mathbin{\curvearrowright} (\chi \mathbin{\curvearrowright} \tilde{\tilde{a}} \cdot x) = \psi \mathbin{\curvearrowright} (\chi \mathbin{\curvearrowright} \tilde{\tilde{a}} \cdot (({}^\bullet\psi \wedge \chi)^\circ \mathbin{\curvearrowright} x)) \;.$$

Axiom HST5 is indispensable in deriving this equation. We can use it, for example, to derive

$$(v = 0) \mathbin{\curvearrowright} (({}^\bullet v + v^\bullet = 1) \mathbin{\curvearrowright} \tilde{\tilde{a}} \cdot \tilde{\tilde{b}})$$
$$= (v = 0) \mathbin{\curvearrowright} (({}^\bullet v + v^\bullet = 1) \mathbin{\curvearrowright} \tilde{\tilde{a}} \cdot ((v = 1) \mathbin{\curvearrowright} \tilde{\tilde{b}})) \;.$$

Using equivalences that are results of real arithmetic, we can, for example, derive the following equation:

$$(v = 0) \mathbin{\curvearrowright} ((\dot{v} = 0) \mathbin{\curvearrowright} \sigma^2_{\mathsf{rel}}((\dot{v} = 1) \mathbin{\curvearrowright} \sigma^3_{\mathsf{rel}}(\tilde{\tilde{a}})))$$
$$= (v = 0) \mathbin{\curvearrowright} ((\dot{v} = 0) \mathbin{\curvearrowright} \sigma^2_{\mathsf{rel}}(\bot)) \;.$$

All processes that can be described by means of the constants and operators of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$, can be described by a basic term. The set $\mathcal{B}$ of *basic terms* is inductively defined by the following rules:

- $\bot \in \mathcal{B}$;
- if $\psi \in {\mathcal{P}_{\mathsf{st}}}^+$, then $\psi \mathbin{\curvearrowright} \tilde{\tilde{\delta}} \in \mathcal{B}$;
- if $\psi \in {\mathcal{P}_{\mathsf{st}}}^+$, $\chi \in {\mathcal{P}_{\mathsf{tr}}}^+$ and $a \in \mathsf{A}$, then $\psi :\to (\chi \mathbin{\curvearrowright} \tilde{\tilde{a}}) \in \mathcal{B}$;
- if $\psi \in {\mathcal{P}_{\mathsf{st}}}^+$, $\chi \in {\mathcal{P}_{\mathsf{tr}}}^+$, $a \in \mathsf{A}$ and $t \in \mathcal{B}$, then $\psi :\to (\chi \mathbin{\curvearrowright} \tilde{\tilde{a}} \cdot t) \in \mathcal{B}$;
- if $\psi, \phi \in {\mathcal{P}_{\mathsf{st}}}^+$, $V \subseteq \mathsf{V}$, $r \in \mathbb{R}^>$ and $t \in \mathcal{B}$, then $\psi :\to (\phi \mathbin{\curvearrowright_V} \sigma^r_{\mathsf{rel}}(t)) \in \mathcal{B}$;
- if $t, t' \in \mathcal{B}$, then $t + t' \in \mathcal{B}$.

Here we write $\mathcal{P}_{\mathsf{st}}{}^+$ and $\mathcal{P}_{\mathsf{tr}}{}^+$ for the restrictions of $\mathcal{P}_{\mathsf{st}}$ and $\mathcal{P}_{\mathsf{tr}}$, respectively, to satisfiable propositions. We can prove that all closed terms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ can be reduced to a basic term.

Theorem 1 (Elimination) *For all closed terms t of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ there exists a basic term t' such that $t = t'$ is derivable from the axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$.*

PROOF. See Appendix A.1. □

If we replace in the third and fourth rule of the definition of $\mathcal{B}$ given above $\psi :\rightarrow (\chi \mathbin{⇂} \tilde{\tilde{a}}) \in \mathcal{B}$ by $\chi \mathbin{⇂} \tilde{\tilde{a}} \in \mathcal{B}$ and $\psi :\rightarrow (\chi \mathbin{⇂} \tilde{\tilde{a}} \cdot t) \in \mathcal{B}$ by $\chi \mathbin{⇂} \tilde{\tilde{a}} \cdot t \in \mathcal{B}$, we still have this result. Even if we add in the fourth rule the condition on χ that χ° implies the signal of t, we still have this result.

We can distinguish two interesting kinds of basic terms. The set $\mathcal{B}^\nu$ of *undelayable* basic terms is inductively defined by the following rules:

- $\bot \in \mathcal{B}^\nu$;
- if $\psi \in \mathcal{P}_{\mathsf{st}}{}^+$, then $\psi \mathbin{↾} \tilde{\tilde{\delta}} \in \mathcal{B}^\nu$;
- if $\psi \in \mathcal{P}_{\mathsf{st}}{}^+$, $\chi \in \mathcal{P}_{\mathsf{tr}}{}^+$ and $a \in \mathsf{A}$, then $\psi :\rightarrow (\chi \mathbin{⇂} \tilde{\tilde{a}}) \in \mathcal{B}^\nu$;
- if $\psi \in \mathcal{P}_{\mathsf{st}}{}^+$, $\chi \in \mathcal{P}_{\mathsf{tr}}{}^+$, $a \in \mathsf{A}$ and $t \in \mathcal{B}$, then $\psi :\rightarrow (\chi \mathbin{⇂} \tilde{\tilde{a}} \cdot t) \in \mathcal{B}^\nu$;
- if $t, t' \in \mathcal{B}^\nu$, then $t + t' \in \mathcal{B}^\nu$.

The set $\mathcal{B}^\sigma$ of *delayable* basic terms is inductively defined by the following rules:

- if $\psi, \phi \in \mathcal{P}_{\mathsf{st}}{}^+$, $V \subseteq \mathsf{V}$, $r \in \mathbb{R}^>$ and $t \in \mathcal{B}$, then $\psi :\rightarrow (\phi \mathbin{⇂}_V \sigma_{\mathsf{rel}}^r(t)) \in \mathcal{B}^\sigma$;
- if $t, t' \in \mathcal{B}^\sigma$, then $t + t' \in \mathcal{B}^\sigma$.

We can prove the following lemmas.

Lemma 2 (Urgency) *For all $t \in \mathcal{B}^\nu$, $\nu_{\mathsf{rel}}(t) = t$ is derivable from the axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$.*

PROOF. Easy, by induction on the structure of undelayable basic term t. □

Lemma 3 (Representation) *For all basic terms t, either $t \in \mathcal{B}^\nu$ or there exists a term $t' \in \mathcal{B}^\nu$ and a term $t'' \in \mathcal{B}^\sigma$ such that $t = t' + t''$ is derivable from the axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$.*

PROOF. Easy, by induction on the structure of basic term t. □

As a corollary of Lemmas 2 and 3, we have the following.

Corollary 4 (Representation) *For all closed terms t of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, either $t = \nu_{\mathsf{rel}}(t)$ is derivable from the axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ or there exists a basic term t' of the form $\sum_{i \in I} \psi_i :\rightarrow (\phi_i \mathbin{\urcorner}_{V_i} \sigma_{\mathsf{rel}}^{r_i}(t_i))$ such that $t = \nu_{\mathsf{rel}}(t) + t'$ is derivable from the axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$.*

2.5 Lifting Rules of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$

Below, we introduce some rules which allow results from real analysis to be lifted to equations about processes.

We assume a mathematical theory MT that includes real arithmetic and real analysis to derive properties of signal evolutions. It is assumed that the state variables and the constants and arithmetic operators of real arithmetic can be used in MT to construct expressions designating real-valued functions of $\mathbb{R}^{\geq}$. Likewise, it is assumed that the relational operators of real arithmetic and the logical constants and connectives can be used in MT to construct expressions designating truth-valued functions of $\mathbb{R}^{\geq}$. It is also assumed that MT is based on the following interpretation of the state variables:

- each state variable v is interpreted as a real-valued function of $\mathbb{R}^{\geq}$ that is piecewise of class C^∞ in $\mathbb{R}^{\geq}$; [1]
- the interpretation of a state variable $\dot{v}$ is the right-hand derivative of the interpretation of the state variable v. [2]

It is further assumed that MT is based on the following interpretation of the constants and arithmetic operators of real arithmetic:

- in expressions designating real numbers, constants and arithmetic operators of real arithmetic are interpreted as usual;
- in expressions designating real-valued functions of $\mathbb{R}^{\geq}$, constants and arithmetic operators of real arithmetic are interpreted as the pointwise extensions of their usual interpretations.

Likewise, it is assumed that MT is based on the following interpretation of

[1] A function $f : I \to \mathbb{R}$, where I is an interval in $\mathbb{R}^{\geq}$, is of class C^∞ in I if $f^{(n)}$, the nth order derivative of f, exists at every point of I, and is continuous on I, for every n; and f is piecewise of class C^∞ in I if I can be partitioned into a finite set $\mathcal{I}$ of left-closed and right-open intervals such that, for each interval $I' \in \mathcal{I}$, the restriction of f to I' is of class C^∞ in I'.

[2] For each function that is piecewise of class C^∞ in some interval I, the right-hand derivative equals the derivative at all points of I where the latter exists.

Table 4
Lifting rules for $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ ($a \in \mathsf{A}_\delta$, $r, s > 0$)

$$\frac{V \subseteq C^\infty[0,r] \vdash_{\mathrm{MT}} \psi(0) \to \forall t \in [0,r] \bullet \phi(t) \leftrightarrow \phi'(t)}{\psi \mathbin{\barwedge} (\phi \mathbin{\barwedge_V} \sigma^r_{\mathsf{rel}}(x)) = \psi \mathbin{\barwedge} (\phi' \mathbin{\barwedge_V} \sigma^r_{\mathsf{rel}}(x))} \quad \text{HSELR1}$$

$$\frac{V \subseteq C^\infty[0,r] \vdash_{\mathrm{MT}} \psi(0) \wedge (\forall t \in [0,r] \bullet \phi(t)) \to \psi'(r)}{\psi \mathbin{\barwedge} (\phi \mathbin{\barwedge_V} \sigma^r_{\mathsf{rel}}(x)) = \psi \mathbin{\barwedge} (\phi \mathbin{\barwedge_V} \sigma^r_{\mathsf{rel}}(\psi' \mathbin{\barwedge} x))} \quad \text{HSELR2}$$

$$\frac{V \subseteq C^\infty[0,r] \vdash_{\mathrm{MT}} \psi(0) \wedge (\forall t \in [0,s] \bullet \phi(t)) \wedge (\exists t \in (s,r] \bullet \forall t' \in (s,t] \bullet \neg\phi(t'))}{\psi \mathbin{\barwedge} (\phi \mathbin{\barwedge_V} \sigma^r_{\mathsf{rel}}(x)) = \psi \mathbin{\barwedge} (\phi \mathbin{\barwedge_V} \sigma^s_{\mathsf{rel}}(\tilde{\tilde{\delta}}))} \quad \text{HSELR3}$$

the relational operators of real arithmetic and the logical constants and connectives:

- in expressions designating truth-values, relational operators of real arithmetic, logical constants and logical connectives are interpreted as usual;
- in expressions designating truth-valued functions of $\mathbb{R}^{\geq}$, relational operators of real arithmetic, logical constants and logical connectives are interpreted as the pointwise extensions of their usual interpretations.

Moreover, it is assumed that the following equivalences concerning the operators ${}^\circ\_$, $\_^\circ$, ${}^\bullet\_$ and $\_^\bullet$ are derivable in MT:

$$\begin{aligned}
&{}^\bullet\psi \Leftrightarrow \psi[{}^\bullet\mathsf{V}/\mathsf{V}] \;,\\
&\psi^\bullet \Leftrightarrow \psi[\mathsf{V}^\bullet/\mathsf{V}] \;,\\
&{}^\circ\chi \Leftrightarrow \exists r_1, \ldots, r_{2n} \in \mathbb{R} \bullet \chi[\mathsf{V}/{}^\bullet\mathsf{V}][r_1, \ldots, r_{2n}/{v_1}^\bullet, \ldots, {v_n}^\bullet, {\dot{v}_1}^\bullet, \ldots, {\dot{v}_n}^\bullet] \;,\\
&\chi^\circ \Leftrightarrow \exists r_1, \ldots, r_{2n} \in \mathbb{R} \bullet \chi[\mathsf{V}/\mathsf{V}^\bullet][r_1, \ldots, r_{2n}/{}^\bullet v_1, \ldots, {}^\bullet v_n, {}^\bullet \dot{v}_1, \ldots, {}^\bullet \dot{v}_n] \;.
\end{aligned}$$

Recall that we use the notations $[{}^\bullet\mathsf{V}/\mathsf{V}]$ and $[\mathsf{V}^\bullet/\mathsf{V}]$ for the replacement of the occurrences of $v \in \mathsf{V} \cup \dot{\mathsf{V}}$ by ${}^\bullet v$ and $v^\bullet$, respectively. We use the notations $[\mathsf{V}/{}^\bullet\mathsf{V}]$ and $[\mathsf{V}/\mathsf{V}^\bullet]$ for the reverse replacements.

The rules for lifting results from real analysis to equations about processes are given in Table 4. We use the notation $\vdash_{\mathrm{MT}}$ for derivability in MT; and we write $V \subseteq C^\infty[0,r]$ to indicate that for each $v \in V$ the restriction of v to $[0,r]$ is of class C^∞ in $[0,r]$.

Lifting rule HSELR1 can, for example, be used to derive

$$((v = 0) \wedge (\dot{v} = 1)) \mathbin{\barwedge} \sigma^1_{\mathsf{rel}}(\tilde{\tilde{a}}) = \mathsf{F} \mathbin{\barwedge} \sigma^1_{\mathsf{rel}}(\tilde{\tilde{a}}) = \bot \;.$$

Note that we cannot derive

$$((v = 0) \wedge (\dot{v} = 1)) \mathbin{\barwedge} \tilde{\tilde{a}} = \mathsf{F} \mathbin{\barwedge} \tilde{\tilde{a}} \;.$$

This is to be expected: although it is impossible that the emitted signal of a process evolves according to $(v = 0) \land (\dot{v} = 1)$, it is possible that a process emits the signal $(v = 0) \land (\dot{v} = 1)$.

Lifting rules HSELR2 and HSELR3 are indispensable in deriving the following equations:

$$\begin{aligned}
&(v = 0) \mathbin{\leftthreetimes} ((v \leq 5 \land \dot{v} = 1) \mathbin{\rightthreetimes} \sigma^4_{\mathsf{rel}}(\tilde{\tilde{a}})) \\
&\quad = (v = 0) \mathbin{\leftthreetimes} ((v \leq 5 \land \dot{v} = 1) \mathbin{\rightthreetimes} \sigma^4_{\mathsf{rel}}((v = 4) \mathbin{\leftthreetimes} \tilde{\tilde{a}})) \,, \\
&(v = 0) \mathbin{\leftthreetimes} ((v \leq 5 \land \dot{v} = 1) \mathbin{\rightthreetimes} \sigma^6_{\mathsf{rel}}(\tilde{\tilde{a}})) \\
&\quad = (v = 0) \mathbin{\leftthreetimes} ((v \leq 5 \land \dot{v} = 1) \mathbin{\rightthreetimes} \sigma^5_{\mathsf{rel}}(\tilde{\tilde{\delta}})) \,.
\end{aligned}$$

The use of signal evolution, as well as the use of signal transition, will be further illustrated in Section 3.3, after we have considered the addition of integration and recursion in Sections 3.1 and 3.2, and in Sections 4.6 and 4.7, after we have considered the addition of parallel composition and encapsulation in Section 4.2.

We will henceforth write $PA \vdash e$ to indicate that equation e is derivable from the axioms and lifting rules of process algebra PA using standard equational reasoning.

Because there exist equations that are only derivable with the help of real analysis, by way of the lifting rules, there is no effective procedure for determining of an arbitrary equation whether it is derivable. Therefore, efficient proof techniques are important. Restrictions that make an effective procedure possible could be useful as well.

If we replace C^∞ by C^1 in the current section and the next one, the results of Section 2.7 go through. In other words, we could have chosen for state variables that are functions from $\mathbb{R}^{\geq}$ to $\mathbb{R}$ that are piecewise of class C^1 in $\mathbb{R}^{\geq}$.[3] However, that choice would complicate the theory and might inhibit useful extensions.

[3] A function $f : I \to \mathbb{R}$, where I is an interval in $\mathbb{R}^{\geq}$, is of class C^1 in I if $\dot{f}$, the (1st order) derivative of f, exists at every point of I, and is continuous on I; and f is piecewise of class C^1 in I if I can be partitioned into a finite set $\mathcal{I}$ of left-closed and right-open intervals such that, for each interval $I' \in \mathcal{I}$, the restriction of f to I' is of class C^1 in I'.

The structural operational semantics of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ will be described below using assignments of state variables. An *assignment* of state variables is a function $\alpha : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ or a function $\beta : {}^\bullet\mathsf{V} \cup \mathsf{V}^\bullet \to \mathbb{R}$. An assignment $\alpha : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ is also called a *state*. An assignment $\beta : {}^\bullet\mathsf{V} \cup \mathsf{V}^\bullet \to \mathbb{R}$ is also called a *state update*.

An assignment $\alpha : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ can be extended to state expressions and atomic state propositions in the usual homomorphic way, and an assignment $\beta : {}^\bullet\mathsf{V} \cup \mathsf{V}^\bullet \to \mathbb{R}$ can be extended to transition expressions and atomic transition propositions in the usual homomorphic way. An assignment $\alpha : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ can also be extended further to state propositions as usual, except for state propositions of the forms $^\circ\chi$ and χ°, and an assignment $\beta : {}^\bullet\mathsf{V} \cup \mathsf{V}^\bullet \to \mathbb{R}$ can also be extended further to transition propositions as usual, except for transition propositions of the forms ${}^\bullet\psi$ and $\psi^\bullet$. We will use the same name for an assignment and its extensions. For state propositions of the forms $^\circ\chi$ and χ°, we have:

- $\alpha(^\circ\chi) = \mathsf{T}$ iff there exists a state update β such that $\beta({}^\bullet v) = \alpha(v)$ for all $v \in \mathsf{V} \cup \dot{\mathsf{V}}$ and $\beta(\chi) = \mathsf{T}$;
- $\alpha(\chi^\circ) = \mathsf{T}$ iff there exists a state update β such that $\beta(v^\bullet) = \alpha(v)$ for all $v \in \mathsf{V} \cup \dot{\mathsf{V}}$ and $\beta(\chi) = \mathsf{T}$.

For transition propositions of the forms ${}^\bullet\psi$ and $\psi^\bullet$, we have:

- $\beta({}^\bullet\psi) = \mathsf{T}$ iff there exists a state α such that $\alpha(v) = \beta({}^\bullet v)$ for all $v \in \mathsf{V} \cup \dot{\mathsf{V}}$ and $\alpha(\psi) = \mathsf{T}$;
- $\beta(\psi^\bullet) = \mathsf{T}$ iff there exists a state α such that $\alpha(v) = \beta(v^\bullet)$ for all $v \in \mathsf{V} \cup \dot{\mathsf{V}}$ and $\alpha(\psi) = \mathsf{T}$.

In Ref. [12], the structural operational semantics of $\mathrm{BPA}_{\mathrm{ps}}$ is described using valuations of atomic propositions. A *valuation* of atomic propositions is a function $v : \mathsf{P}_{\mathsf{at}} \to \mathbb{B}$. In the case of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, where the set of atomic state propositions is taken as the set P_{at} of atomic propositions, an assignment $\alpha : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ of state variables induces a valuation $\alpha : \mathsf{P}_{\mathsf{at}} \to \mathbb{B}$ of atomic propositions, viz. the extension of the assignment α to atomic state propositions.

Below, satisfaction of state propositions by state evolutions and satisfaction of transition propositions by pairs of states will be defined. Let $\rho : [0, r] \to (\mathsf{V} \to \mathbb{R})$, where $r \in \mathbb{R}^>$, and $V \subseteq \mathsf{V}$. Then, for every $v \in \mathsf{V}$, we write ρ_v for the function $\rho_v : [0, r] \to \mathbb{R}$ defined by $\rho_v(t) = \rho(t)(v)$. We say that ρ is a *state evolution* if ρ_v is piecewise of class C^∞ in $[0, r)$ for all $v \in \mathsf{V}$. If ρ is a state evolution, we say that ρ is *smooth* for V if ρ_v is of class C^∞ in $[0, r]$ for all $v \in V$. If ρ is a state evolution, we say that a state α *agrees* with ρ at time t,

$t \in [0,r]$, if for all $v \in \mathsf{V}$:

$$\alpha(v) = \rho_v(t) \;, \quad \alpha(\dot{v}) = \dot{\rho}_v(t) \;.$$

Let (α, α') be a pair of states. Then, we say that a state update β *agrees* with (α, α') if for all $v \in \mathsf{V}$:

$$\beta({}^\bullet v) = \alpha(v) \;, \quad \beta({}^\bullet \dot{v}) = \alpha(\dot{v}) \;, \quad \beta(v^\bullet) = \alpha'(v) \;, \quad \beta(\dot{v}^\bullet) = \alpha'(\dot{v}) \;.$$

We write $\mathcal{E}_r$ for the set of all state evolutions $\rho : [0,r] \to (\mathsf{V} \to \mathbb{R})$. For a given state evolution $\rho : [0,r] \to (\mathsf{V} \to \mathbb{R})$ and a given time $t \in [0,r]$, there is a unique state that agrees with ρ at t. We write α_t^ρ for this unique state. For a given pair of states (α, α'), there is a unique state update that agrees with (α, α'). We write $\beta_{\alpha'}^{\alpha}$ for this unique state update.

Satisfaction of state propositions (by states and state evolutions) and satisfaction of transition propositions (by state transitions) are used below in describing the structural operational semantics of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$. Satisfaction of state propositions and transition propositions is defined as follows:

- a state proposition ψ is *satisfied* by state α, written $\alpha \models \psi$, if

  $$\alpha(\psi) = \mathsf{T} \;;$$

- a state proposition ϕ is *satisfied* by state evolution $\rho \in \mathcal{E}_r$, written $\rho \models \phi$, if

  $$\alpha_t^\rho(\phi) = \mathsf{T} \text{ for all } t \in [0,r] \;;$$

- a transition proposition χ is *satisfied* by the transition from state α to state α', written $\alpha \rightarrowtail \alpha' \models \chi$, if

  $$\beta_{\alpha'}^{\alpha}(\chi) = \mathsf{T} \;.$$

We write $\alpha \overset{r,\rho}{\longmapsto} \alpha' \models_V \phi$ for

$$\rho \in \mathcal{E}_r,\ \alpha_0^\rho = \alpha,\ \alpha_r^\rho = \alpha',\ \rho \text{ is smooth for } V \text{ and } \rho \models \phi \;.$$

Note that we have for all states α and α':

$$\begin{aligned}
&\alpha \models {}^\circ({}^\bullet\psi) && \text{iff } \alpha \models (\psi^\bullet)^\circ \text{ iff } \alpha \models \psi \ ,\\
&\alpha \models {}^\circ(\psi^\bullet) && \text{iff } \alpha \models ({}^\bullet\psi)^\circ \text{ iff there exists a state } \alpha'' \text{ such that } \alpha'' \models \psi \ ,\\
&\alpha \rightarrow \alpha' \models {}^\bullet\psi && \text{iff } \alpha \models \psi \ ,\\
&\alpha \rightarrow \alpha' \models \psi^\bullet && \text{iff } \alpha' \models \psi \ ,\\
&\alpha \rightarrow \alpha' \models \chi && \text{implies } \alpha \models {}^\circ\chi \text{ and } \alpha' \models \chi^\circ \ ,\\
&\alpha \stackrel{r,\rho}{\longmapsto} \alpha' \models_V \phi && \text{implies } \alpha \models \phi \text{ and } \alpha' \models \phi \ .
\end{aligned}$$

The structural operational semantics of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ is described by the rules given in Tables 5 and 6. In Table 6, we use a to stand for elements of A_δ. The following transition relations are used:

- a binary relation $\langle \_, \alpha\rangle \xrightarrow{a} \langle \_, \alpha'\rangle$ for each $a \in \mathsf{A}$, $\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$;
- a unary relation $\langle \_, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$ for each $a \in \mathsf{A}$, $\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$;
- a binary relation $\langle \_, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle \_, \alpha'\rangle$ for each $r \in \mathbb{R}^>$, $\rho \in \mathcal{E}_r$, $\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ such that $\alpha = \alpha_0^\rho$ and $\alpha' = \alpha_r^\rho$;
- a unary relation $\alpha \in [\mathsf{s}(\_)]$ for each $\alpha : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$.

We write $\langle t, \alpha\rangle \not\stackrel{\surd}{\longmapsto}$ for the set of all transition formulas $\neg(\langle t, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t', \alpha'\rangle)$ where t' is a closed term of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, $\alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$ and $\rho \in \mathcal{E}_r$. We write $\rho \trianglerighteq r$, where $\rho \in \mathcal{E}_{r+s}$ $(r, s > 0)$, for the $\rho' \in \mathcal{E}_s$ such that $\rho'(s') = \rho(r + s')$ for all $s' \in [0, s]$.

The four kinds of transition relations are called the *action step*, *action termination*, *time step* and *signal* relations, respectively. They can be explained as follows:

- $\langle t, \alpha\rangle \xrightarrow{a} \langle t', \alpha'\rangle$: in state α, process t is capable of first performing action a at the current point of time and then proceeding as process t' in state α';
- $\langle t, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$: in state α, process t is capable of first performing action a at the current point of time and then terminating successfully in state α';
- $\langle t, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t', \alpha'\rangle$: in state α, process t is capable of first idling for a period of time r, while the state evolves according to ρ, and then proceeding as process t' in state α';
- $\alpha \in [\mathsf{s}(t)]$: in state α, the signal emitted by process t holds.

The following are important properties of the transition relations defined by the rules given in Tables 5 and 6. We have for all closed terms t and t', for all

Table 5
Rules for operational semantics of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ ($a \in \mathsf{A}$, $r, s > 0$)

$$\frac{}{\langle \tilde{\tilde{a}}, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle \sigma^0_{\mathsf{rel}}(x), \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle} \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle \sigma^0_{\mathsf{rel}}(x), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle} \quad \frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}{\langle \sigma^0_{\mathsf{rel}}(x), \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}$$

$$\frac{}{\langle \sigma^{r+s}_{\mathsf{rel}}(x), \alpha\rangle \overset{r,\rho}{\longmapsto} \langle \sigma^s_{\mathsf{rel}}(x), \alpha'\rangle} \quad \frac{\alpha' \in [\mathsf{s}(x)]}{\langle \sigma^r_{\mathsf{rel}}(x), \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x, \alpha'\rangle} \quad \frac{\langle x, \alpha'\rangle \overset{s,\rho \trianglerighteq r}{\longmapsto} \langle x', \alpha''\rangle}{\langle \sigma^r_{\mathsf{rel}}(x), \alpha\rangle \overset{r+s,\rho}{\longmapsto} \langle x', \alpha''\rangle}$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle,\ \alpha \in [\mathsf{s}(y)]}{\langle x + y, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle} \quad \frac{\alpha \in [\mathsf{s}(x)],\ \langle y, \alpha\rangle \xrightarrow{a} \langle y', \alpha'\rangle}{\langle x + y, \alpha\rangle \xrightarrow{a} \langle y', \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle,\ \alpha \in [\mathsf{s}(y)]}{\langle x + y, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle} \quad \frac{\alpha \in [\mathsf{s}(x)],\ \langle y, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle x + y, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle,\ \langle y, \alpha\rangle \overset{r}{\not\longmapsto},\ \alpha \in [\mathsf{s}(y)]}{\langle x + y, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle} \quad \frac{\langle x, \alpha\rangle \overset{r}{\not\longmapsto},\ \alpha \in [\mathsf{s}(x)],\ \langle y, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle y', \alpha'\rangle}{\langle x + y, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle y', \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle,\ \langle y, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle y', \alpha'\rangle}{\langle x + y, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x' + y', \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle x \cdot y, \alpha\rangle \xrightarrow{a} \langle x' \cdot y, \alpha'\rangle} \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle,\ \alpha' \in [\mathsf{s}(y)]}{\langle x \cdot y, \alpha\rangle \xrightarrow{a} \langle y, \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}{\langle x \cdot y, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x' \cdot y, \alpha'\rangle}$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle \psi :\rightharpoonup x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}\ \alpha \models \psi \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle \psi :\rightharpoonup x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}\ \alpha \models \psi$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}{\langle \psi :\rightharpoonup x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}\ \alpha \models \psi$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle \psi \mathbin{\triangle\!\!\blacktriangleright} x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}\ \alpha \models \psi \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle \psi \mathbin{\triangle\!\!\blacktriangleright} x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}\ \alpha \models \psi$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}{\langle \psi \mathbin{\triangle\!\!\blacktriangleright} x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}\ \alpha \models \psi$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle \phi \mathbin{\triangledown\!\!\blacktriangleright}_V x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}\ \alpha \models \phi \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle \phi \mathbin{\triangledown\!\!\blacktriangleright}_V x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}\ \alpha \models \phi$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}{\langle \phi \mathbin{\triangledown\!\!\blacktriangleright}_V x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle \phi \mathbin{\triangledown\!\!\blacktriangleright}_V x', \alpha'\rangle}\ \alpha \overset{r,\rho}{\longmapsto} \alpha' \models_V \phi$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle \chi \mathbin{\triangledown\!\!\blacktriangleright} x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}\ \alpha \rightarrow \alpha' \models \chi \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle \chi \mathbin{\triangledown\!\!\blacktriangleright} x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}\ \alpha \rightarrow \alpha' \models \chi$$

$$\frac{\langle x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}{\langle \chi \mathbin{\triangledown\!\!\blacktriangleright} x, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle x', \alpha'\rangle}\ \alpha \models {}^\circ\chi$$

$$\frac{\langle x, \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}{\langle \nu_{\mathsf{rel}}(x), \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle} \quad \frac{\langle x, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}{\langle \nu_{\mathsf{rel}}(x), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}$$

Table 6
Rules for $\alpha \in [\mathsf{s}(\_)]$ ($a \in \mathsf{A}_\delta$, $r > 0$)

$$\frac{}{\alpha \in [\mathsf{s}(\tilde{a})]} \quad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\sigma^0_{\mathsf{rel}}(x))]} \quad \frac{}{\alpha \in [\mathsf{s}(\sigma^r_{\mathsf{rel}}(x))]} \quad \frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)]}{\alpha \in [\mathsf{s}(x + y)]} \quad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(x \cdot y)]}$$

$$\frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\psi :\rightarrow x)]} \quad \frac{}{\alpha \in [\mathsf{s}(\psi :\rightarrow x)]}\ \alpha \not\models \psi \quad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\psi \between x)]}\ \alpha \models \psi$$

$$\frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\phi \between_V x)]}\ \alpha \models \phi \quad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\chi \between x)]} \quad \frac{}{\alpha \in [\mathsf{s}(\chi \between x)]}\ \alpha \not\models {}^\circ\chi \quad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\nu_{\mathsf{rel}}(x))]}$$

$\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$, $a \in \mathsf{A}$, $r \in \mathbb{R}^>$ and $\rho \in \mathcal{E}_r$:

$$\langle t, \alpha\rangle \xrightarrow{a} \langle t', \alpha'\rangle \text{ or } \langle t, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle \text{ or } \langle t, \alpha\rangle \xmapsto{r,\rho} \langle t', \alpha'\rangle \text{ implies } \alpha \in [\mathsf{s}(t)]\ ,$$

$$\langle t, \alpha\rangle \xrightarrow{a} \langle t', \alpha'\rangle \text{ or } \langle t, \alpha\rangle \xmapsto{r,\rho} \langle t', \alpha'\rangle \text{ implies } \alpha' \in [\mathsf{s}(t')]\ .$$

In work on hybrid automata, the transition systems associated with hybrid automata usually include time step relations $\langle \_, \alpha\rangle \xmapsto{r} \langle \_, \alpha'\rangle$ instead of $\langle \_, \alpha\rangle \xmapsto{r,\rho} \langle \_, \alpha'\rangle$. State evolutions only play a part as "witnesses" for time steps, see e.g. Ref. [3]. Time step relations $\langle \_, \alpha\rangle \xmapsto{r} \langle \_, \alpha'\rangle$ would yield a semantics which is too abstract for our purpose. For instance, the meaning of $\phi \between_V (\phi' \between_{V'} t)$ would be far from its intended meaning, and axiom HSE11 would not be sound. Consider, for example, the following terms:

$$(x = 0 \wedge y = 0) \between (((x + y \leq 4 \wedge x \geq y \wedge \neg(x = 1 \wedge y = 1) \wedge \dot{x} > 0 \wedge \dot{y} > 0) \between \sigma^*_{\mathsf{rel}}((x = 2 \wedge y = 2) :\rightarrow \widetilde{\widetilde{stop}}))\ ,$$

$$(x = 0 \wedge y = 0) \between (((x + y \leq 4 \wedge x \leq y \wedge \dot{x} > 0 \wedge \dot{y} > 0) \between ((x + y \leq 4 \wedge x \geq y \wedge \neg(x = 1 \wedge y = 1) \wedge \dot{x} > 0 \wedge \dot{y} > 0) \between \sigma^*_{\mathsf{rel}}((x = 2 \wedge y = 2) :\rightarrow \widetilde{\widetilde{stop}})))\ .$$ [4]

The first term can be regarded as describing an object that first moves smoothly from point $(0, 0)$ to point $(2, 2)$, staying away from the left of the straight line through points $(0, 0)$ and $(2, 2)$, and not going through point $(1, 1)$, and then stops. According to our intuition, the second term expresses that the object is on top of that staying away from the right of the straight line through points $(0, 0)$ and $(2, 2)$. This is impossible and therefore the object will never stop. As to be expected, it is derivable from the axioms and lifting rules of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$

[4] The notation $\sigma^*_{\mathsf{rel}}(t)$, which will be introduced in Section 3.1, is to be read as "the relative delay of t for an arbitrary period of time".

that the second term equals:

$$(x = 0 \wedge y = 0) \trianglerighteq$$
$$((x + y \leq 4 \wedge x = y \wedge \neg(x = 1 \wedge y = 1) \wedge \dot{x} > 0 \wedge \dot{y} > 0) \trianglerighteq \sigma^*_{\mathsf{rel}}(\tilde{\tilde{\delta}})) \; .$$

There are movements satisfying the first restriction and movements satisfying the second restriction. However, in the case where the state evolutions representing those movements are only playing a part as witnesses in the operational semantics, it is kept unnoticed that the two restrictions cannot be satisfied both. As a result, the second term would not denote a process that will never stop. With regard to the first term, note that the velocity of the object must change as time goes by in order to meet the constraints on its position. However, no discontinuities are allowed.

2.7 Bisimulation and Soundness

Bisimulation based on the transition rules for $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ is defined as usual in cases where processes with different states are not considered to be equivalent.

A *bisimulation* is a symmetric binary relation B on pairs of closed terms and states, called *configurations*, such that for all configurations $\langle t_1, \alpha\rangle, \langle t_2, \alpha\rangle$ with $B(\langle t_1, \alpha\rangle, \langle t_2, \alpha\rangle)$ the following conditions hold:

- whenever $\langle t_1, \alpha\rangle \xrightarrow{a} \langle t_1', \alpha'\rangle$, then there is a t_2' such that $\langle t_2, \alpha\rangle \xrightarrow{a} \langle t_2', \alpha'\rangle$ and $B(\langle t_1', \alpha'\rangle, \langle t_2', \alpha'\rangle)$;
- whenever $\langle t_1, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$, then $\langle t_2, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$;
- whenever $\langle t_1, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle t_1', \alpha'\rangle$, then there is a t_2' such that $\langle t_2, \alpha\rangle \overset{r,\rho}{\longmapsto} \langle t_2', \alpha'\rangle$ and $B(\langle t_1', \alpha'\rangle, \langle t_2', \alpha'\rangle)$;
- whenever $\alpha \in [\mathsf{s}(t_1)]$, then $\alpha \in [\mathsf{s}(t_2)]$.

Two configurations $\langle t_1, \alpha_1\rangle$ and $\langle t_2, \alpha_2\rangle$ are *bisimulation equivalent* (or simply *bisimilar*), written $\langle t_1, \alpha_1\rangle \leftrightarroweq \langle t_2, \alpha_2\rangle$, if $\alpha_1 = \alpha_2$ and there exists a bisimulation B such that $B(\langle t_1, \alpha_1\rangle, \langle t_2, \alpha_2\rangle)$. Two closed terms t_1 and t_2 are *bisimulation equivalent*, written $t_1 \leftrightarroweq t_2$, if $\langle t_1, \alpha\rangle \leftrightarroweq \langle t_2, \alpha\rangle$ for all states α.

We also consider a variant of bisimulation equivalence, called interference-compatible bisimulation equivalence, which is finer than bisimulation equivalence. The idea behind interference-compatible bisimulation is the following. A process proceeding in parallel with a process P can change the state of P at any time. Interference-compatible bisimulation offers resistance to such changes. For example, if a configuration $\langle t_1, \alpha\rangle$ is related to a configuration $\langle t_2, \alpha\rangle$ and $\langle t_1, \alpha\rangle \xrightarrow{a} \langle t_1', \alpha'\rangle$, then there is a t_2' such that $\langle t_2, \alpha\rangle \xrightarrow{a} \langle t_2', \alpha'\rangle$ and $\langle t_1', \alpha''\rangle$ is related to $\langle t_2', \alpha''\rangle$ for all states α''. Parallel composition is in-

troduced in Section 4. In that section, the need for interference-compatible bisimulation equivalence will be explained.

An *interference-compatible bisimulation* is a symmetric binary relation B on closed terms such that for all closed terms t_1, t_2 with $B(t_1, t_2)$ the following conditions hold:

- whenever $\langle t_1, \alpha\rangle \xrightarrow{a} \langle t_1', \alpha'\rangle$, then there is a t_2' such that $\langle t_2, \alpha\rangle \xrightarrow{a} \langle t_2', \alpha'\rangle$ and $B(t_1', t_2')$;
- whenever $\langle t_1, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$, then $\langle t_2, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$;
- whenever $\langle t_1, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t_1', \alpha'\rangle$, then there is a t_2' such that $\langle t_2, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t_2', \alpha'\rangle$ and $B(t_1', t_2')$;
- whenever $\alpha \in [\mathsf{s}(t_1)]$, then $\alpha \in [\mathsf{s}(t_2)]$.

Two closed terms t_1 and t_2 are *interference-compatible bisimulation equivalent*, written $t_1 \leftrightarrows t_2$, if there exists an interference-compatible bisimulation B such that $B(t_1, t_2)$. We will use *ic-bisimulation* as an abbreviation for interference-compatible bisimulation.

We regard ic-bisimulation equivalence less natural than bisimulation equivalence: it appears to waver between two opinions. Besides, both axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3 are not sound under ic-bisimulation equivalence.

Bisimulation equivalence is coarser than ic-bisimulation equivalence.

Lemma 5 (Inclusion) *For all closed terms t_1 and t_2, if $t_1 \leftrightarrows t_2$ then $t_1 \leftrightarrow t_2$.*

PROOF. Suppose that $t_1 \leftrightarrows t_2$. Suppose further that B is an ic-bisimulation witnessing that $t_1 \leftrightarrows t_2$. Define $B' = \{(\langle t_1, \alpha\rangle, \langle t_2, \alpha\rangle) \mid B(t_1, t_2), \alpha \text{ is a state}\}$. It is easy to see that B' is a bisimulation. Moreover, if $B'(\langle t_1, \alpha\rangle, \langle t_2, \alpha\rangle)$, then $B'(\langle t_1, \alpha'\rangle, \langle t_2, \alpha'\rangle)$ for all states α'. So $t_1 \leftrightarrow t_2$. $\Box$

Bisimulation equivalence and ic-bisimulation equivalence are preserved by all operators of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$.

Theorem 6 (Congruence) *Both bisimulation equivalence and ic-bisimulation equivalence are congruences with respect to the operators of* $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$.

PROOF. See Appendix A.2. $\Box$

The axioms and lifting rules of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ are sound with respect to bisimulation equivalence.

Theorem 7 (Soundness) *For all closed terms* t_1 *and* t_2 *of* $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$*, we have* $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}} \vdash t_1 = t_2$ *implies* $t_1 \leftrightarrows t_2$*.*

PROOF. See Appendix A.3. □

We shall henceforth use the name $\mathrm{icBPA}_{\mathrm{hs}}^{\mathrm{srt}}$ to refer to the process algebra that differs from $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ only by the absence of axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3. As a corollary of the proof of Theorem 7, we have the following.

Corollary 8 (Soundness) *For all closed terms* t_1 *and* t_2 *of* $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$*, we have* $\mathrm{icBPA}_{\mathrm{hs}}^{\mathrm{srt}} \vdash t_1 = t_2$ *implies* $t_1 \leftrightarrows\!\!\!= t_2$*.*

Bisimulation equivalence appears to be preferable. After all, axiom HST5 is indispensable to analyse how signal transition changes the signal of a process over performing an action and lifting rules HSELR2 and HSELR3 are indispensable to analyse how signal evolution changes the signal of a process during idling. Axiom HST14 is a simple alternative to axiom HST5, which has its limitation.

3 Integration and Recursion

In this section, we extend $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ with integration and guarded recursion. These extensions will be needed in many applications. We illustrate this by means of an example concerning a thermostat. We also pay some attention to Zeno behaviour, which can be described in $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ extended with integration and guarded recursion.

3.1 $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ *with Integration*

In order to cover processes that are capable of performing an action at all points in a certain time interval, we add integration to $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$. Integration is represented by the variable-binding operator $\int$. Let P be an expression, possibly containing variable u, such that $P[p/u]$ (P with p substituted for u) represents a process for all $p \in \mathbb{R}^{\geq}$; and let $U \subseteq \mathbb{R}^{\geq}$. Then the *integration* $\int_{u \in U} P$ behaves like one of the processes $P[p/u]$ for $p \in U$. Hence, integration is a form of alternative composition over a set of alternatives that may even be a continuum.

Table 7
Axioms for integration ($p \geq 0$)

$\int_{u \in U} F(u) = \int_{u' \in U} F(u')$	INT1
$\int_{u \in \emptyset} F(u) = \tilde{\tilde{\delta}}$	INT2
$\int_{u \in \{p\}} F(u) = F(p)$	INT3
$\int_{u \in U \cup U'} F(u) = \int_{u \in U} F(u) + \int_{u \in U'} F(u)$	INT4
$U \neq \emptyset \Rightarrow \int_{u \in U} x = x$	INT5
$(\forall u \in U \bullet F(u) = G(u)) \Rightarrow \int_{u \in U} F(u) = \int_{u \in U} G(u)$	INT6
U, U' unbounded $\Rightarrow \int_{u \in U} \sigma^u_{\mathsf{rel}}(\tilde{\tilde{\delta}}) = \int_{u \in U'} \sigma^u_{\mathsf{rel}}(\tilde{\tilde{\delta}})$	INT8SR
$\sup U = p,\ p \in U \Rightarrow \int_{u \in U} \sigma^u_{\mathsf{rel}}(\tilde{\tilde{\delta}}) = \sigma^p_{\mathsf{rel}}(\tilde{\tilde{\delta}})$	INT9SR
$\int_{u \in U} \sigma^p_{\mathsf{rel}}(F(u)) = \sigma^p_{\mathsf{rel}}(\int_{u \in U} F(u))$	INT10SR
$\int_{u \in U} (F(u) + G(u)) = \int_{u \in U} F(u) + \int_{u \in U} G(u)$	INT11
$\int_{u \in U} (F(u) \cdot x) = (\int_{u \in U} F(u)) \cdot x$	INT12
$\int_{u \in U} \nu_{\mathsf{rel}}(F(u)) = \nu_{\mathsf{rel}}(\int_{u \in U} F(u))$	INT13
$\int_{u \in U} (\psi :\rightarrow F(u)) = \psi :\rightarrow \int_{u \in U} F(u)$	PSINT1
$\int_{u \in U} (\psi \mathbin{\triangle} F(u)) = \psi \mathbin{\triangle} \int_{u \in U} F(u)$	PSINT2
$\int_{u \in U} (\phi \mathbin{\triangledown_V} F(u)) = \phi \mathbin{\triangledown_V} \int_{u \in U} F(u)$	HSINT1
$\int_{u \in U} (\chi \mathbin{\triangledown} F(u)) = \chi \mathbin{\triangledown} \int_{u \in U} F(u)$	HSINT2

We shall henceforth use F and G as variables ranging over functions that map each non-negative real number to a process and can be represented by terms containing a designated free variable ranging over $\mathbb{R}^{\geq}$. For more information on such second-order variables, see e.g. Refs. [35,36]. Furthermore, we shall henceforth use $u, u', \ldots$ as variables ranging over $\mathbb{R}^{\geq}$. It is assumed that each first-order definable set of non-negative real numbers can be denoted by a closed term, and we shall henceforth use $U, U', \ldots$ to stand for arbitrary closed terms denoting first-order definable sets of non-negative real numbers.

The additional axioms for integration are the equations given in Table 7. Axiom INT1 is similar to the α-conversion rule of λ-calculus. Axioms INT2–INT4 show that integration is a form of alternative composition over a set of alternatives. Axiom INT5 can be regarded as the counterpart of axiom A3 for integration. Axiom INT6 is an extensionality axiom. The remaining axioms are easily understood by realizing that integration is a form of alternative composition over a set of alternatives. Axioms INT10SR, INT11, INT12, INT13, PSINT1, PSINT2, HSINT1 and HSINT2 can simply be regarded as variants of axioms SRT3, A2, A4, SRU3, GC4, SE3, HSE7 and HST7, respectively. Axioms INT8SR and INT9SR are both reminiscent of the equation $\sigma^{p+q}_{\mathsf{rel}}(\tilde{\tilde{\delta}}) + \sigma^p_{\mathsf{rel}}(\tilde{\tilde{\delta}}) = \sigma^{p+q}_{\mathsf{rel}}(\tilde{\tilde{\delta}})$, which is derivable from axioms A6SR, SRT2 and SRT3.

Table 8
Additional rules for integration ($a \in \mathsf{A}$, $p, q \geq 0$, $r > 0$)

$$\frac{\langle F(p), \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle,\ \{\alpha \in [\mathsf{s}(F(q))] \mid q \in U\}}{\langle \int_{u\in U} F(u), \alpha\rangle \xrightarrow{a} \langle x', \alpha'\rangle}\ p \in U$$

$$\frac{\langle F(p), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle,\ \{\alpha \in [\mathsf{s}(F(q))] \mid q \in U\}}{\langle \int_{u\in U} F(u), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle}\ p \in U$$

$$\frac{\begin{array}{c}\{\langle F(q), \alpha\rangle \overset{r,\rho}{\longmapsto} \langle F_1(q), \alpha'\rangle \mid q \in U_1\},\\ \ldots,\\ \{\langle F(q), \alpha\rangle \overset{r,\rho}{\longmapsto} \langle F_n(q), \alpha'\rangle \mid q \in U_n\},\\ \{\langle F(q), \alpha\rangle \overset{r}{\not\longmapsto},\ \alpha \in [\mathsf{s}(F(q))] \mid q \in U_{n+1}\}\end{array}}{\langle \int_{u\in U} F(u), \alpha\rangle \overset{r,\rho}{\longmapsto} \langle \int_{u\in U_1} F_1(u) + \ldots + \int_{u\in U_n} F_n(u), \alpha'\rangle}\ \begin{array}{l}\{U_1, \ldots, U_n\} \text{ partition}\\ \text{of } U \setminus U_{n+1}, U_{n+1} \subset U\end{array}$$

$$\frac{\{\alpha \in [\mathsf{s}(F(q))] \mid q \in U\}}{\alpha \in [\mathsf{s}(\int_{u\in U} F(u))]}$$

The following important equation concerning the interchange of integration order is derivable:

$$\int_{u\in U}(\int_{u'\in U'} K(u, u')) = \int_{u'\in U'}(\int_{u\in U} K(u, u'))\ .$$

Like F and G, we use K here as a variable ranging over functions that map each pair of non-negative real numbers to a process and can be represented by terms containing a pair of designated free variables ranging over $\mathbb{R}^{\geq}$.

The additional axioms for integration in the case of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ are essentially the additional axioms for integration in the case of $\mathrm{BPA}^{\mathrm{srt}}$ and on top of that axioms concerning the interaction of integration with conditional proceeding, signal emission, signal evolution and signal transition.

We shall henceforth use the name $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT to refer to the extension of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ with integration.

We shall henceforth use the notation $\sigma^*_{\mathsf{rel}}(t)$ for $\int_{u\in[0,\infty)} \sigma^u_{\mathsf{rel}}(t)$, with u a variable not occurring free in t.

The structural operational semantics for integration is described by the rules given in Table 8. The complexity of the rule concerning the time-related capabilities of a process $\int_{u\in U} F(u)$ is caused by the fact that the processes $F(p)$ with $p \in U$ that are capable of idling need not change uniformly while idling. For more information on this phenomenon, see e.g. Refs. [11,26]. Bisimulation equivalence and ic-bisimulation equivalence are preserved by integration. All additional axioms for integration are sound with respect to bisimulation equivalence and ic-bisimulation equivalence.

Table 9
Additional axioms for guarded recursion

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

3.2 $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ *with Guarded Recursion*

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

A *recursive specification* over $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ is a set of *recursive* equations $E = \{X = t_X \mid X \in V\}$ where V is a set of variables and each t_X is a term of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ 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 $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$) $\{P_X \mid X \in \mathrm{V}(E)\}$ 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 $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ containing a variable X. We call an occurrence of X in t *guarded* if t has a subterm of the form $\tilde{\tilde{a}} \cdot t'$ or $\sigma_{\mathsf{rel}}^r(t')$, where $a \in \mathsf{A}$, $r > 0$ and t' a term of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, with t' containing this occurrence of X. A recursive specification over $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ 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 $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ and the equations of the recursive specification. A guarded recursive specification has a unique solution.

For each guarded recursive specification E and each variable $X \in \mathrm{V}(E)$, we introduce a constant $\langle X|E\rangle$ which is interpreted as the unique solution of E for X. 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 $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ with guarded recursion and E be a guarded recursive specification. Then we write $\langle t|E\rangle$ for t with, for all $X \in \mathrm{V}(E)$, all occurrences of X in t replaced by $\langle X|E\rangle$.

We shall henceforth use $X, Y, \ldots$ as variables ranging over processes in the case where they occur in a recursive specification. Furthermore, we shall henceforth use $t_X, t_Y, \ldots$ to stand for arbitrary terms of which the closed substitution instances denote processes, and $E, E', \ldots$ to stand for arbitrary guarded recursive specifications.

The additional axioms for guarded recursion are the equations given in Table 9. 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

Table 10
Additional rules for guarded recursion ($a \in \mathsf{A}$, $r > 0$)

$$\frac{\langle\langle t_X|E\rangle,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle}{\langle\langle X|E\rangle,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle} \; X = t_X \in E \qquad \frac{\langle\langle t_X|E\rangle,\alpha\rangle \xrightarrow{a} \langle\surd,\alpha'\rangle}{\langle\langle X|E\rangle,\alpha\rangle \xrightarrow{a} \langle\surd,\alpha'\rangle} \; X = t_X \in E$$

$$\frac{\langle\langle t_X|E\rangle,\alpha\rangle \xmapsto{r,\rho} \langle x',\alpha'\rangle}{\langle\langle X|E\rangle,\alpha\rangle \xmapsto{r,\rho} \langle x',\alpha'\rangle} \; X = t_X \in E \qquad \frac{\alpha \in [\mathsf{s}(\langle t_X|E\rangle)]}{\alpha \in [\mathsf{s}(\langle X|E\rangle)]} \; X = t_X \in E$$

guarded 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.

It is sometimes helpful to rewrite guarded recursive specifications. The following useful fact about the rewriting of guarded recursive specifications can be proven. Let E and E' be two guarded recursive specifications over $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$, where E' is E rewritten using the axioms of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ and the equations of E. Then the equation $\langle X|E\rangle = \langle X|E'\rangle$ is derivable for all $X \in \mathrm{V}(E)$.

The additional axioms for guarded recursion in the case of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ are the same as in the cases of BPA, $\mathrm{BPA}_{\mathrm{ps}}$ and $\mathrm{BPA}^{\mathrm{srt}}$. Guarded recursion is added in the same way to $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT and the other extensions of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ presented in this paper.

We shall henceforth use the name $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC to refer to the extension of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT with guarded recursion.

The structural operational semantics for guarded recursion is described by the rules given in Table 10. Bisimulation equivalence and ic-bisimulation equivalence are preserved by guarded recursion. All additional axioms for guarded recursion are sound with respect to bisimulation equivalence and ic-bisimulation equivalence.

3.3 Example: Thermostat

In this section, we consider a thermostat. We give a guarded recursive specification of the behaviour of the thermostat. This example is adapted from Ref. [7]. We take the following (adapted) informal description of the behaviour of the thermostat from Ref. [37] as the starting point of our specification.

Initially, the temperature is $18\,^\circ\mathrm{C}$ and the heating is on. While the heating is on, the temperature T in the room goes up according to the differential equation $\dot{T} = -T + 22$. When the temperature becomes $20\,^\circ\mathrm{C}$, the heating will be turned off. While the heating is off, the temperature T in the room

goes down according to the the differential equation $\dot{T} = -T + 17$. When the temperature becomes $18\,^\circ$C, the heating will be turned on again.

The recursive specification of the thermostat consists of the following equations:

$$\begin{aligned}
Th &= (T = 18) \trianglerighteq Th^{\mathrm{on}} \,, \\
Th^{\mathrm{on}} &= (18 \leq T \leq 20 \wedge \dot{T} = -T + 22) \trianglerighteq \\
&\qquad \sigma^*_{\mathsf{rel}}\Big((T = 20) \mathbin{:\!\rightarrow} \Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\textit{turn-off}} \cdot Th^{\mathrm{off}}\Big)\Big) \,, \\
Th^{\mathrm{off}} &= (18 \leq T \leq 20 \wedge \dot{T} = -T + 17) \trianglerighteq \\
&\qquad \sigma^*_{\mathsf{rel}}\Big((T = 18) \mathbin{:\!\rightarrow} \Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\textit{turn-on}} \cdot Th^{\mathrm{on}}\Big)\Big) \,.
\end{aligned}$$

The signal transition operator $\trianglerighteq$ and the signal evolution operator $\trianglerighteq$ are needed here to make precise that the temperature in the room does not change instantaneously at the points of time at which the heating is turned off or on and that the temperature in the room changes continuously as described above during the periods in between.

Using the axioms and lifting rules of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC, we can prove that the solution of this recursive specification is the same as the solution of the recursive specification that consists of the following equations:

$$\begin{aligned}
Th' &= (T = 18) \trianglerighteq Th'^{\mathrm{on}} \,, \\
Th'^{\mathrm{on}} &= (18 \leq T \leq 20 \wedge \dot{T} = -T + 22) \trianglerighteq \\
&\qquad \sigma^{\ln 2}_{\mathsf{rel}}\Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\textit{turn-off}} \cdot Th'^{\mathrm{off}}\Big) \,, \\
Th'^{\mathrm{off}} &= (18 \leq T \leq 20 \wedge \dot{T} = -T + 17) \trianglerighteq \\
&\qquad \sigma^{\ln 3}_{\mathsf{rel}}\Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\textit{turn-on}} \cdot Th'^{\mathrm{on}}\Big) \,.
\end{aligned}$$

It is clear from this specification that the heater is on for a fraction $\ln 2/\ln 3$ of the time. If we could hide the atomic propositions concerning the state variable T, we would even get the process recursively specificied by the following equation:

$$Th'' = \sigma^{\ln 2}_{\mathsf{rel}}\Big(\widetilde{\textit{turn-off}}\Big) \cdot \sigma^{\ln 3}_{\mathsf{rel}}\Big(\widetilde{\textit{turn-on}}\Big) \cdot Th'' \,.$$

For properties that do not concern the course of the values of T and $\dot{T}$, the processes Th' and Th'' do not show a single difference. Therefore, we would like to add a hiding operator $v\,\Delta$ for each $v \in \mathsf{V}$ such that $v\,\Delta\,P$ is the

process that behaves like P, but with the dependence of its behaviour on the value of v and $\dot{v}$ made invisible. With the envisaged operator, we would have $T \Delta Th' = Th''$. However, this extension would require a semantics that carries more detail than the structural operational semantics given in this paper. For that reason, we consider it a topic for future work.

3.4 Zeno Behaviour

Consider an object that moves on a flat plane as follows. It starts moving from the point $(1, 0)$, i.e. the point with x-coordinate 1 and y-coordinate 0, such that $\dot{x} = -1$ and $\dot{y} = 0.5$. When the x-coordinate becomes 0, it proceeds moving such that $\dot{x} = 0.5$ and $\dot{y} = -1$. When the y-coordinate becomes 0 once more, it proceeds moving again such that $\dot{x} = -1$ and $\dot{y} = 0.5$. And so on, and so forth. Thus the object approaches in a zig-zag way the point $(0, 0)$, but never reaches it. Moreover, the direction of the object changes infinitely many times before 2 time units have elapsed. This phenomenon, infinitely many instantaneous state changes happening in a non-zero finite amount of time, is called Zeno behaviour. Obviously, such behaviour is unrealizable.

Nevertheless, $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$+INT+REC is expressive enough to describe Zeno behaviour. For example, the behaviour considered above can be described by the following equations:

$$\begin{aligned}
O &= (x = 1 \mathbin{\wedge} y = 0) \mathbin{\triangle} O^r \,, \\
O^r &= (x \geq 0 \mathbin{\wedge} \dot{x} = -1 \mathbin{\wedge} \dot{y} = 0.5) \mathbin{\triangledown} \\
&\qquad \sigma^*_{\mathsf{rel}}\Big((x = 0) :\!\rightarrow \Big((x^\bullet = {}^\bullet x \mathbin{\wedge} y^\bullet = {}^\bullet y) \mathbin{\triangledown} \widetilde{\textit{turn-left}} \cdot O^l\Big)\Big) \,, \\
O^l &= (y \geq 0 \mathbin{\wedge} \dot{x} = 0.5 \mathbin{\wedge} \dot{y} = -1) \mathbin{\triangledown} \\
&\qquad \sigma^*_{\mathsf{rel}}\Big((y = 0) :\!\rightarrow \Big((x^\bullet = {}^\bullet x \mathbin{\wedge} y^\bullet = {}^\bullet y) \mathbin{\triangledown} \widetilde{\textit{turn-right}} \cdot O^r\Big)\Big) \,.
\end{aligned}$$

Under bisimulation equivalence and ic-bisimulation equivalence, no distinction is made between behaviours that occur after a point of time at which infinitely many instantaneous state changes accumulate.

4 Algebra of Communicating Processes

In this section, we extend $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ with operators to capture parallelism and communication. Beforehand, we give already an idea of the application of the resulting process algebra, called $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, by means of an example concerning

the temperature control of a nuclear reactor. We also illustrate its application by means of examples concerning a bottle filling system and a railroad crossing system.

4.1 Example: Nuclear Reactor

This section is a sample of the application of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. It is meant to give a first impression of how one describes the behaviour of hybrid systems composed of several components that proceed concurrently and interact with each other using $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. We describe the behaviour of a simple nuclear reactor in which the temperature of the reactor core is controlled by two control rods. This example is adapted from Ref. [6]. We take the following informal description of the behaviour of the reactor as the starting point of our formal description.

Initially, the temperature of the reactor core is $510\,^\circ\mathrm{C}$ and the control rods are outside the reactor core. With the control rods outside the reactor core, the temperature T increases according to the differential equation $\dot{T} = 0.1T - 50$. The reactor must be shut down if the temperature becomes higher than $550\,^\circ\mathrm{C}$. To prevent a shutdown, one of the control rods should be put into the reactor core once the temperature becomes $550\,^\circ\mathrm{C}$. With control rod 1 inside the reactor core, the temperature T decreases according to the differential equation $\dot{T} = 0.1T - 56$. With control rod 2 inside the reactor core, the temperature T decreases according to the differential equation $\dot{T} = 0.1T - 60$. The control rod inside the reactor is removed from the reactor core once the temperature becomes $510\,^\circ\mathrm{C}$. When it is removed, it cannot be put back in the reactor core for the next c seconds. To prevent that the reactor ever needs to be shut down, the time c must be short enough to guarantee that, whenever the temperature of the reactor core becomes $550\,^\circ\mathrm{C}$, one of the control rods can be put back in the reactor core.

The recursive specification of the reactor core consists of the following equa-

tions:

$$
\begin{aligned}
C &= (T = 510) \trianglerighteq C^{\mathrm{out}} \,, \\
C^{\mathrm{out}} &= (T \leq 550 \wedge \dot{T} = 0.1T - 50) \trianglerighteq \\
&\quad \Big(\sigma^*_{\mathsf{rel}}\Big((T = 550) :\rightarrow \Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\widetilde{s_1(add)}} \cdot C^{\mathrm{in1}}\Big)\Big) \\
&\quad\quad + \sigma^*_{\mathsf{rel}}\Big((T = 550) :\rightarrow \Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\widetilde{s_2(add)}} \cdot C^{\mathrm{in2}}\Big)\Big)\Big) \,, \\
C^{\mathrm{in1}} &= (T \geq 510 \wedge \dot{T} = 0.1T - 56) \trianglerighteq \\
&\quad \sigma^*_{\mathsf{rel}}\Big((T = 510) :\rightarrow \Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\widetilde{s_1(rmv)}} \cdot C^{\mathrm{out}}\Big)\Big) \,, \\
C^{\mathrm{in2}} &= (T \geq 510 \wedge \dot{T} = 0.1T - 60) \trianglerighteq \\
&\quad \sigma^*_{\mathsf{rel}}\Big((T = 510) :\rightarrow \Big((T^\bullet = {}^\bullet T) \trianglerighteq \widetilde{\widetilde{s_2(rmv)}} \cdot C^{\mathrm{out}}\Big)\Big) \,.
\end{aligned}
$$

Each of the control rods is recursively defined by a single equation:

$$
\begin{aligned}
R_1 &= \sigma^*_{\mathsf{rel}}\Big(\widetilde{\widetilde{r_1(add)}} \cdot \sigma^*_{\mathsf{rel}}\Big(\widetilde{\widetilde{r_1(rmv)}} \cdot \sigma^c_{\mathsf{rel}}(R_1)\Big)\Big) \,, \\
R_2 &= \sigma^*_{\mathsf{rel}}\Big(\widetilde{\widetilde{r_2(add)}} \cdot \sigma^*_{\mathsf{rel}}\Big(\widetilde{\widetilde{r_2(rmv)}} \cdot \sigma^c_{\mathsf{rel}}(R_2)\Big)\Big) \,.
\end{aligned}
$$

Assuming that the whole system starts with both control rods out of the core for at least c seconds, the reactor is described by the following term:

$$\partial_H(C \parallel R_1 \parallel R_2) \,,$$

where

$$
\begin{aligned}
H = {} & \{s_i(d) \mid i \in \{1,2\}, d \in \{add, rmv\}\} \\
& \cup \{r_i(d) \mid i \in \{1,2\}, d \in \{add, rmv\}\} \,.
\end{aligned}
$$

We write $s_i(d)$, $r_i(d)$ and $c_i(d)$ for the action of sending d at port i, the action of receiving d at port i and the action of communicating d at port i, respectively. The action $c_i(d)$ is the action that is left when $s_i(d)$ and $r_i(d)$ are performed synchronously. This notation is the standardized notation for handshaking communication introduced for ACP in Ref. [38].

At this stage, we cannot explain this description fully. However, note that it appears to be a fairly direct representation of the informal description given above. In addition to constants and operators of $\mathrm{ACP}^{\mathrm{srt}}$ [11] and $\mathrm{ACP}_{\mathrm{ps}}$ [12], the signal transition operator $\trianglerighteq$ and the signal evolution operator $\trianglerighteq$ introduced in Section 2.4 are used. These operators are needed to make precise that the

temperature of the reactor core does not change instantaneously at the points of time at which a control rod is put into it or removed from it and that the temperature of the reactor core changes continuously as described above during the periods in between.

4.2 ACP$^{\mathrm{srt}}$ *for Hybrid Systems*

The basic ways of combining atomic processes into composite processes are sequential and alternative composition. A more advanced way of combining processes is parallel composition. Let P_1 and P_2 be processes. Then the *parallel composition* of P_1 and P_2, written $P_1 \parallel P_2$, is the process that proceeds with P_1 and P_2 in parallel. By this is roughly meant that it can behave in the following ways:

- first either P_1 or P_2 performs its first action and next it proceeds in parallel with the process following that action and the process that did not perform an action;
- if their first actions can be performed synchronously, first P_1 and P_2 perform their first actions synchronously and next it proceeds in parallel with the processes following those actions.

However, P_1 and P_2 may have to idle before they can perform their first action. Therefore, their parallel composition can only start with:

- performing an action of P_1 or P_2 if it can do so before or at the ultimate point of time for the other process to start performing actions or to deadlock;
- performing an action of P_1 and an action of P_2 synchronously if both processes can do so at the same point of time.

Moreover, the state transition caused by performing the first action of P_1 or P_2 must be one that is not precluded by the other process. By this is meant that:

- the signal of the other process must hold in the state immediately before the transition and the state immediately after the transition;
- if the other process is idling when the action is performed, a state evolution with discontinuities for all state variables of which the value changes by the transition must be possible for the other process.

We say that the discontinuities resulting from the transition are possible for the other process to indicate that the latter condition is fulfilled.

The point of view is that there is only one action left when actions are performed synchronously. Thus, we can amongst other things easily model hand-

shaking communication: when the action $s_i(d)$ of sending datum d at port i and the action $r_i(d)$ of receiving datum d at port i are performed synchronously, only the action $c_i(d)$ of communicating datum d at port i is left.

Parallel composition does not prevent actions that can be performed synchronously from being performed on their own. In order to capture parallelism and communication fully, we have, in addition to parallel composition, encapsulation with respect to a certain set of actions. Let P be a process and H be a set of actions. Then the *encapsulation* of P with respect to H, written $\partial_H(P)$, keeps P from performing actions in H. The process P becomes deadlocked at the point that one of these actions would otherwise be performed. The name encapsulation is used here because the actions in H are encapsulated from communication with actions coming from the environment of P.

We will use two auxiliary operators in the axiomatization of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$: $\between$ and $|$. The operator $\between$ is interpreted as left merge, which is the same as parallel composition except that the left merge of P_1 and P_2 can only start with performing an action of P_1. The operator $|$ is interpreted as communication merge, which is the same as parallel composition except that the communication merge of P_1 and P_2 can only start with performing an action of P_1 and an action of P_2 synchronously.

We shall henceforth use $H, H', \ldots$ to stand for arbitrary subsets of A.

It is assumed that a fixed but arbitrary partial commutative and associative *communication* function $\gamma : \mathsf{A} \times \mathsf{A} \to \mathsf{A}$ has been given. The function γ is regarded to give the result of synchronously performing any two actions for which this is possible, and to be undefined otherwise.

The additional axioms for parallel composition and encapsulation are the equations given in Tables 11, 12, 13, 14 and 15. Adding the equations given in Table 11 to the axioms of $\mathrm{BPA}_{\bot}^{\mathrm{srt}}$ gives us the subtheory $\mathrm{ACP}_{\bot}^{\mathrm{srt}}$, $\mathrm{ACP}^{\mathrm{srt}}$ with non-existence. Adding the equations given in Tables 11 and 12 to the axioms of $\mathrm{BPA}_{\mathrm{ps}}^{\mathrm{srt}}$ gives us the subtheory $\mathrm{ACP}_{\mathrm{ps}}^{\mathrm{srt}}$, $\mathrm{ACP}^{\mathrm{srt}}$ with propositional signals. Adding the equations given in Tables 11, 12, 13 and 14, with the exception of axioms CM2SRPS and CM3SRPS from Table 11, to the axioms and lifting rule of $\mathrm{icBPA}_{\mathrm{hs}}^{\mathrm{srt}}$ gives us the theory $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, $\mathrm{ACP}^{\mathrm{srt}}$ for hybrid systems (note that axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3 are not present in $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$). Adding the same equations, together with the equations given in Table 15, to the axioms and lifting rule of $\mathrm{icBPA}_{\mathrm{hs}}^{\mathrm{srt}}$ with integration gives us the theory $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ with integration.

First of all, we look at the additional axioms for $\mathrm{ACP}_{\bot}^{\mathrm{srt}}$ (Table 11). Axioms CM1, CM4, CM8, CM9, D3 and D4 are in common with ACP. Axioms CM2SRPS, CM3SRPS, CM5SR–CM7SR, CF1SR, CF2SR, D1SR and D2SR are simple reformulations of axioms CM2, CM3, CM5–CM7, CF1, CF2, D1

Table 11
Additional axioms for $\mathrm{ACP}_{\bot}^{\mathrm{srt}}$ ($a, b, c \in \mathsf{A}_\delta$, $p \geq 0$, $r > 0$)

$x \parallel y = x \between y + y \between x + x \mid y$	CM1
$\tilde{\tilde{a}} \between x = \tilde{\tilde{a}} \cdot x + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$	CM2SRPS
$\tilde{\tilde{a}} \cdot x \between y = \tilde{\tilde{a}} \cdot (x \parallel y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	CM3SRPS
$\sigma^r_{\mathsf{rel}}(x) \between \nu_{\mathsf{rel}}(y) = \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	SRCM1aPS
$\sigma^r_{\mathsf{rel}}(x) \between (\nu_{\mathsf{rel}}(y) + z) = \sigma^r_{\mathsf{rel}}(x) \between z + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	SRCM1bPS
$\sigma^r_{\mathsf{rel}}(x) \between \sigma^r_{\mathsf{rel}}(y) = \sigma^r_{\mathsf{rel}}(x \between y)$	SRCM2
$(x + y) \between z = x \between z + y \between z$	CM4
$\tilde{\tilde{a}} \cdot x \mid \tilde{\tilde{b}} = (\tilde{\tilde{a}} \mid \tilde{\tilde{b}}) \cdot x$	CM5SR
$\tilde{\tilde{a}} \mid \tilde{\tilde{b}} \cdot x = (\tilde{\tilde{a}} \mid \tilde{\tilde{b}}) \cdot x$	CM6SR
$\tilde{\tilde{a}} \cdot x \mid \tilde{\tilde{b}} \cdot y = (\tilde{\tilde{a}} \mid \tilde{\tilde{b}}) \cdot (x \parallel y)$	CM7SR
$\nu_{\mathsf{rel}}(x) \mid \sigma^r_{\mathsf{rel}}(y) = \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$	SRCM3PS
$\sigma^r_{\mathsf{rel}}(x) \mid \nu_{\mathsf{rel}}(y) = \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	SRCM4PS
$\sigma^r_{\mathsf{rel}}(x) \mid \sigma^r_{\mathsf{rel}}(y) = \sigma^r_{\mathsf{rel}}(x \mid y)$	SRCM5
$(x + y) \mid z = x \mid z + y \mid z$	CM8
$x \mid (y + z) = x \mid y + x \mid z$	CM9
$\tilde{\tilde{a}} \mid \tilde{\tilde{b}} = \tilde{\tilde{c}}$ if $\gamma(a, b) = c$	CF1SR
$\tilde{\tilde{a}} \mid \tilde{\tilde{b}} = \tilde{\tilde{\delta}}$ if $\gamma(a, b)$ undefined	CF2SR
$\partial_H(\tilde{\tilde{a}}) = \tilde{\tilde{a}}$ if $a \notin H$	D1SR
$\partial_H(\tilde{\tilde{a}}) = \tilde{\tilde{\delta}}$ if $a \in H$	D2SR
$\partial_H(\sigma^p_{\mathsf{rel}}(x)) = \sigma^p_{\mathsf{rel}}(\partial_H(x))$	SRD
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D3
$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$	D4
$\bot \between x = \bot$	NE4
$\bot \mid x = \bot$	NE5
$x \mid \bot = \bot$	NE6
$\partial_H(\bot) = \bot$	NE7

and D2 of ACP. For a detailed introduction to ACP, see Ref. [10]. Axioms SRCM1aPS, SRCM1bPS, SRCM2, SRCM3PS, SRCM4PS, SRCM5 and SRD are new axioms concerning the interaction of relative delay with left merge, communication merge and encapsulation. The axioms given in Table 11, other than axioms NE4–NE7, are the axioms concerning parallel composition and encapsulation of $\mathrm{ACP}^{\mathrm{srt}}$ without the deadlocked process (see Ref. [11]), but with $\tilde{\tilde{\delta}}$ replaced by $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$ or $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$ in the axioms of which the name ends with PS. This is to accommodate the addition of propositional signals: the signal of the left merge and communication merge of two processes is

Table 12
Additional axioms for $\mathrm{ACP}_{\mathrm{ps}}^{\mathrm{srt}}$ $(r > 0)$

$(\psi :\rightarrow x) \between y = \psi :\rightarrow (x \between y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	PSCM1
$(\psi :\rightarrow x) \mid y = \psi :\rightarrow (x \mid y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	PSCM2
$x \mid (\psi :\rightarrow y) = \psi :\rightarrow (x \mid y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$	PSCM3
$(\psi \mathbin{\triangle} x) \between y = \psi \mathbin{\triangle} (x \between y)$	PSCM4
$(\psi \mathbin{\triangle} x) \mid y = \psi \mathbin{\triangle} (x \mid y)$	PSCM5
$x \mid (\psi \mathbin{\triangle} y) = \psi \mathbin{\triangle} (x \mid y)$	PSCM6
$\sigma^r_{\mathsf{rel}}(x) \between (\psi :\rightarrow y + z) =$ $\psi :\rightarrow (\sigma^r_{\mathsf{rel}}(x) \between (y + z)) + \neg\psi :\rightarrow (\sigma^r_{\mathsf{rel}}(x) \between z)$	PSSRCM
$\partial_H(\psi :\rightarrow x) = \psi :\rightarrow \partial_H(x)$	PSD1
$\partial_H(\psi \mathbin{\triangle} x) = \psi \mathbin{\triangle} \partial_H(x)$	PSD2

always the conjunction of the signals of both processes. Axioms NE4–NE7 concern the effect of left merge, communication merge and encapsulation on the non-existent process. The equation $t \between \bot = \bot$ is derivable for all closed terms t. The axioms of $\mathrm{ACP}_{\bot}^{\mathrm{srt}}$ are essentially the axioms of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\bot}$ with on top of that axiom NESRU (Table 1) concerning the effect of relative undelayable time-out on the non-existent process.

Secondly, we look at the additional axioms for $\mathrm{ACP}_{\mathrm{ps}}^{\mathrm{srt}}$ (Table 12). Axioms PSCM1–PSCM6, PSD1 and PSD2 are similar to the additional axioms for $\mathrm{ACP}_{\mathrm{ps}}$ (see Ref. [12]). Terms of the form $\mathsf{s}_\rho(x) \mathbin{\triangle} \delta$ have been replaced by terms of the form $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$ instead of $\mathsf{s}_\rho(x) \mathbin{\triangle} \tilde{\tilde{\delta}}$. However, the addition of the operator s_ρ would yield the derivability of $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t)) = \mathsf{s}_\rho(t) \mathbin{\triangle} \tilde{\tilde{\delta}}$ for all closed terms t. The other differences are due to the absence of the terminal signal emission operator and the choice of having as the signal of the left merge of two processes, as in the case of the communication merge, always the conjunction of the signals of both processes. This choice, originating from the variant with discrete relative timing introduced in Ref. [34], is required for axiom PSSRCM to be sound. Axiom PSSRCM is useful dealing with the parallel composition of processes that are conditionally capable of idling. Note that axioms NE4–NE7 are derivable from axioms PSCM4, PSCM5, PSCM6, PSD2 and SE2. Note further that the following generalization of axiom PSSRCM is derivable:

$$\begin{aligned}
&\sigma^r_{\mathsf{rel}}(x) \between (\psi :\rightarrow y + \psi' :\rightarrow z)\\
&\quad = (\psi \wedge \psi') :\rightarrow (\sigma^r_{\mathsf{rel}}(x) \between (y + z))\\
&\qquad + (\psi \wedge \neg\psi') :\rightarrow (\sigma^r_{\mathsf{rel}}(x) \between y) + (\neg\psi \wedge \psi') :\rightarrow (\sigma^r_{\mathsf{rel}}(x) \between z) \;.
\end{aligned}$$

The following equation is derivable for all closed terms t and t':

$$t \between (\psi \mathbin{\triangle} t') = \psi \mathbin{\triangle} (t \between \psi \mathbin{\triangle} t') \;.$$

Table 13
Additional axioms for $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ ($a \in \mathsf{A}_\delta$, $r, s > 0$)

$\tilde{\tilde{a}} \between x = \mathsf{d}_\rho(x) \rightarrow \tilde{\tilde{a}} \cdot x + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$	CM2SRHS
$\tilde{\tilde{a}} \cdot x \between y = \mathsf{d}_\rho(y) \rightarrow \tilde{\tilde{a}} \cdot (x \parallel y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	CM3SRHS
$(\phi \rightarrow_V x) \between y = \phi \rightarrow_V (x \between y)$	HSCM1
$(\phi \rightarrow_V x) \mid y = \phi \rightarrow_V (x \mid y)$	HSCM2
$x \mid (\phi \rightarrow_V y) = \phi \rightarrow_V (x \mid y)$	HSCM3
$(\chi \rightarrow \nu_{\mathsf{rel}}(x)) \between y = \chi \rightarrow (\nu_{\mathsf{rel}}(x) \between y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	HSCM4
$(\chi \rightarrow x) \mid y = \chi \rightarrow (x \mid y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(y))$	HSCM5
$x \mid (\chi \rightarrow y) = \chi \rightarrow (x \mid y) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$	HSCM6
$\sigma^r_{\mathsf{rel}}(x) \between (\phi \rightarrow_V \sigma^r_{\mathsf{rel}}(y) + z) =$ $\sigma^r_{\mathsf{rel}}(x) \between (\sigma^r_{\mathsf{rel}}(\phi \rightarrow_V y) + z) + \phi \rightarrow_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$	HSSRCM
$\partial_H(\phi \rightarrow_V x) = \phi \rightarrow_V \partial_H(x)$	HSD1
$\partial_H(\chi \rightarrow x) = \chi \rightarrow \partial_H(x)$	HSD2

Table 14
Axioms for root discontinuity operator ($a \in \mathsf{A}_\delta$, $r > 0$)

$\mathsf{d}_\rho(\bot) = \mathsf{F}$	RDO1	$\mathsf{d}_\rho(\psi :\to x) = {}^\bullet\psi \to \mathsf{d}_\rho(x)$	RDO6
$\mathsf{d}_\rho(\tilde{\tilde{a}}) = \mathsf{T}$	RDO2	$\mathsf{d}_\rho(\psi \mathbin{\barwedge} x) = {}^\bullet\psi \wedge \mathsf{d}_\rho(x)$	RDO7
$\mathsf{d}_\rho(\sigma^r_{\mathsf{rel}}(x)) = \mathsf{T}$	RDO3	$\mathsf{d}_\rho(\phi \rightarrow_V \nu_{\mathsf{rel}}(x)) = {}^\bullet\phi \wedge \mathsf{d}_\rho(\nu_{\mathsf{rel}}(x))$	RDO8
$\mathsf{d}_\rho(x + y) = \mathsf{d}_\rho(x) \wedge \mathsf{d}_\rho(y)$	RDO4	$\mathsf{d}_\rho(\phi \rightarrow_V \sigma^r_{\mathsf{rel}}(x)) = {}^\bullet\phi \wedge \mathcal{C}_V$	RDO9
$\mathsf{d}_\rho(x \cdot y) = \mathsf{d}_\rho(x)$	RDO5	$\mathsf{d}_\rho(\chi \rightarrow x) = {}^\bullet(^\circ\chi) \to \mathsf{d}_\rho(x)$	RDO10

The axioms of $\mathrm{ACP}_{\mathrm{ps}}^{\mathrm{srt}}$ are essentially the axioms of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$ with on top of that axiom PSSRCM, axiom NESRU (Table 1) concerning the effect of relative undelayable time-out on the non-existent process and axioms PSSRU1 and PSSRU2 (Table 2) concerning the interaction of relative undelayable time-out with conditional proceeding and signal emission.

Finally, we look at the additional axioms for $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ (Tables 13 and 14). Axioms CM2SRHS and CM3SRHS from Table 13 replace axioms CM2SRPS and CM3SRPS from Table 11. These new axioms are needed to reflect that in the parallel composition of two processes the discontinuities resulting from the transition caused by performing the first action of one of them must be possible for the other. The auxiliary root discontinuity operator d_ρ, of which axioms RDO1–RDO10 are the defining equations, yields the transition proposition that characterizes the transitions from which only discontinuities result that are possible for a process. Recall that $\mathcal{C}_V$ abbreviates $\bigwedge_{v \in V}(v^\bullet = {}^\bullet v \wedge \dot{v}^\bullet = {}^\bullet\dot{v})$. The following substitution instances of axioms CM2SRPS and CM3SRPS are derivable for all closed terms t in which the signal evolution operator only

occurs in subterms of the form $\phi \mathbin{\urcorner}_{\emptyset} t'$:

$$\tilde{\tilde{a}} \between t = \tilde{\tilde{a}} \cdot t + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t)) \;,$$
$$\tilde{\tilde{a}} \cdot x \between t = \tilde{\tilde{a}} \cdot (x \parallel t) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t)) \;.$$

Hence, the auxiliary operator d_ρ, and the replacement of axioms CM2SRPS and CM3SRPS by axioms CM2SRHS and CM3SRHS, would not be needed if the preclusion of discontinuities for certain state variables in state evolutions was not supported. Axioms HSCM1–HSCM6, HSSRCM, HSD1 and HSD2 show that signal evolution and signal transition take effect over what takes place first, also in the presence of parallel composition and encapsulation. Obviously, we do not have $x \between (\phi \mathbin{\urcorner}_V y) = \phi \mathbin{\urcorner}_V (x \between y)$: y may not be done with idling when x performs its first action. Axiom HSSRCM shows that notwithstanding that, if two processes idle in parallel, signal evolution takes place in a way possible for both processes. Note that the following variation of axiom HSSRCM is derivable:

$$\begin{aligned}&\sigma^r_{\mathsf{rel}}(x) \between (\phi \mathbin{\urcorner}_V \sigma^r_{\mathsf{rel}}(y) + \phi' \mathbin{\urcorner}_{V'} \sigma^r_{\mathsf{rel}}(z))\\ &\quad = \sigma^r_{\mathsf{rel}}(x) \between (\sigma^r_{\mathsf{rel}}(\phi \mathbin{\urcorner}_V y) + \sigma^r_{\mathsf{rel}}(\phi' \mathbin{\urcorner}_{V'} z)) + (\phi \wedge \phi') \mathbin{\urcorner}_{V \cup V'} \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}) \;.\end{aligned}$$

Using the axioms of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$, we can, for example, derive the following equations:

$$\begin{aligned}&\sigma^2_{\mathsf{rel}}((v^\bullet = {}^\bullet v + 1) \mathbin{\urcorner} \tilde{\tilde{a}} \cdot \sigma^1_{\mathsf{rel}}(\tilde{\tilde{b}})) \parallel ((\dot{v} = 0) \mathbin{\urcorner}_{\{v\}} \sigma^3_{\mathsf{rel}}(\tilde{\tilde{c}}))\\ &\quad = (\dot{v} = 0) \mathbin{\urcorner}_{\{v\}} \sigma^2_{\mathsf{rel}}(\tilde{\tilde{\delta}}) \;,\end{aligned}$$

$$\begin{aligned}&\sigma^2_{\mathsf{rel}}((v^\bullet = {}^\bullet v + 1) \mathbin{\urcorner} \tilde{\tilde{a}} \cdot \sigma^1_{\mathsf{rel}}(\tilde{\tilde{b}})) \parallel ((\dot{v} = 0) \mathbin{\urcorner}_{\emptyset} \sigma^3_{\mathsf{rel}}(\tilde{\tilde{c}}))\\ &\quad = (\dot{v} = 0) \mathbin{\urcorner}_{\emptyset}\\ &\qquad \sigma^2_{\mathsf{rel}}((v^\bullet = {}^\bullet v + 1) \mathbin{\urcorner} \tilde{\tilde{a}} \cdot ((\dot{v} = 0) \mathbin{\urcorner}_{\emptyset} \sigma^1_{\mathsf{rel}}(\tilde{\tilde{b}} \cdot (\dot{v} = 0) \mathbin{\barwedge} \tilde{\tilde{c}} + \tilde{\tilde{c}} \cdot \tilde{\tilde{b}}))) \;.\end{aligned}$$

Note the difference on the left hand side of these equations: $(\dot{v} = 0) \mathbin{\urcorner}_{\{v\}} \sigma^3_{\mathsf{rel}}(\tilde{\tilde{c}})$ precludes discontinuities for v, but $(\dot{v} = 0) \mathbin{\urcorner}_{\emptyset} \sigma^3_{\mathsf{rel}}(\tilde{\tilde{c}})$ does not preclude discontinuities for v.

We can prove that all closed terms of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$ can be reduced to a closed term of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$.

Theorem 9 (Elimination) *For all closed terms t of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$, there exists a closed term t' of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ such that $t = t'$ is derivable from the axioms of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$.*

PROOF. See Appendix A.4. □

Table 15
Additional axioms for integration

$\int_{u\in U}(F(u) \between x) = (\int_{u\in U}F(u)) \between x$	INT14
$\int_{u\in U}(F(u) \mid x) = (\int_{u\in U}F(u)) \mid x$	INT15
$\int_{u\in U}(x \mid F(u)) = x \mid (\int_{u\in U}F(u))$	INT16
$\int_{u\in U}\partial_H(F(u)) = \partial_H(\int_{u\in U}F(u))$	INT17
$\mathsf{d}_\rho(\int_{u\in U}F(u)) = \bigwedge_{p\in U}\mathsf{d}_\rho(F(p))$	RDO11

As a corollary of Theorem 9, we have that all closed terms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ can be reduced to a basic term.

Corollary 10 (Elimination) *For all closed terms t of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, there exists a basic term t' such that $t = t'$ is derivable from the axioms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$.*

Integration can be added to $\mathrm{ACP}_{\perp}^{\mathrm{srt}}$, $\mathrm{ACP}_{\mathrm{ps}}^{\mathrm{srt}}$ as well as $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. The additional axioms for integration (Table 15) can be regarded as variants of axioms CM4, CM8, CM9, D3 and RDO4.

We shall henceforth use the name $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$+INT to refer to the extension of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ with integration and the name $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$+INT+REC to refer to the extension of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$+INT with guarded recursion.

4.3 Two-Phase Derivation

The equations added to the axioms of $\mathrm{icBPA}_{\mathrm{hs}}^{\mathrm{srt}}$ to obtain $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, cannot be added to $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$: if axiom HST5, axiom HST14, lifting rule HSELR2 or lifting rule HSELR3 is added to $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, the result is not sound. For example, we can derive the following equation from the axioms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$:

$$\begin{aligned}&((v=0) \trianglelefteq ((v^\bullet = {}^\bullet v + 1) \trianglerighteq \tilde{\tilde{a}} \cdot ((v=0) \trianglerighteq \tilde{\tilde{b}}))) \between ((v^\bullet = {}^\bullet v - 1) \trianglerighteq \tilde{\tilde{c}})\\ &\quad = (v=0) \trianglelefteq ((v^\bullet = {}^\bullet v + 1) \trianglerighteq \tilde{\tilde{a}} \cdot ((v^\bullet = {}^\bullet v - 1) \trianglerighteq \tilde{\tilde{c}} \cdot \tilde{\tilde{b}}))\ .\end{aligned}$$

However, if we add axiom HST5 to $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, we can also derive the following equation:

$$\begin{aligned}&(v=0) \trianglelefteq ((v^\bullet = {}^\bullet v + 1) \trianglerighteq \tilde{\tilde{a}} \cdot ((v=0) \trianglerighteq \tilde{\tilde{b}}))\\ &\quad = (v=0) \trianglelefteq ((v^\bullet = {}^\bullet v + 1) \trianglerighteq \tilde{\tilde{a}} \cdot \tilde{\tilde{\delta}})\ .\end{aligned}$$

Then by substitution of the right-hand side for the left-hand side in the previous equation, and next further derivation from the axioms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, we

get:

$$\begin{aligned}&((v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot ((v = 0) \mathbin{:\!\rightarrow} \tilde{\tilde{b}}))) \between ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}})\\ &\quad = ((v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot \tilde{\tilde{\delta}})) \between ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}})\\ &\quad = (v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}} \cdot \tilde{\tilde{\delta}}))\ .\end{aligned}$$

Yet, we have that

$$\begin{aligned}&(v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}} \cdot \tilde{\tilde{b}}))\\ &\quad \not\leftrightarrows (v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}} \cdot \tilde{\tilde{\delta}}))\ ,\end{aligned}$$

$$\begin{aligned}&(v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}} \cdot \tilde{\tilde{b}}))\\ &\quad \not\leftrightarrows (v = 0) \mathbin{\leftthreetimes} ((v^\bullet = {}^\bullet v + 1) \mathbin{\triangleright} \tilde{\tilde{a}} \cdot ((v^\bullet = {}^\bullet v - 1) \mathbin{\triangleright} \tilde{\tilde{c}} \cdot \tilde{\tilde{\delta}}))\ .\end{aligned}$$

The problem is that bisimulation equivalence is not preserved by parallel composition, left merge and communication merge whereas axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3 are not sound under ic-bisimulation equivalence. Because we still want to use these axioms and lifting rules, we introduce two-phase derivation which only permits the use of axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3 in the absence of parallel composition, left merge and communication merge. With the introduction of two-phase derivation we follow an idea from Ref. [39], where this kind of derivation was introduced to deal with a comparable problem.

Let t_1 and t_2 be closed terms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. Then $t_1 = t_2$ is *two-phase derivable* from the axioms and lifting rule of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ and the axioms and lifting rules of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$, written $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}/\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}} \vdash_2 t_1 = t_2$, if there exist closed terms t_1' and t_2' of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ such that

$$\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}} \vdash t_1 = t_1',\ \mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}} \vdash t_2 = t_2',\ \mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}} \vdash t_1' = t_2'\ .$$

Let t_1 and t_2 be closed terms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$+Ext, where Ext is INT, INT+REC or INT+REC+HSL. Then $t_1 = t_2$ is *two-phase derivable* from the axioms and lifting rule of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$+Ext and the axioms and lifting rules of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$+Ext, written $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}/\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}} + \mathrm{Ext} \vdash_2 t_1 = t_2$ if there exist closed terms t_1' and t_2' of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$+Ext such that

$$\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}} + \mathrm{Ext} \vdash t_1 = t_1',\ \mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}} + \mathrm{Ext} \vdash t_2 = t_2',\ \mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}} + \mathrm{Ext} \vdash t_1' = t_2'\ .$$

Here, HSL refers to the extension with localization, which is treated in Section 5.

It is worth mentioning that the proofs of Theorems 1 and 9 show that axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3 are not needed to obtain the elimination results. If they would be needed, the idea of two-phase derivation would be useless.

Two-phase derivation does not permit the undesirable derivation given above. However, it does permit the derivations leading to the simplications of descriptions of hybrid systems shown in Sections 4.6 and 4.7. Those simplications, which facilitate analysis of the systems concerned, would not be possible otherwise.

The need for two-phase derivation originates from the potentiality of interference between parallel processes through shared state variables. Two-phase derivation may hinder a modular approach to hybrid system description and analysis. To remedy this largely, we could adapt two-phase derivation in such a way that it takes into account the absence of shared state variables.

4.4 Operational Semantics of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$

The structural operational semantics for parallel composition, left merge, communication merge and encapsulation is described by the rules given in Tables 16 and 17. In Table 17, we use a to stand for elements of A_δ. The following additional transition relations are used:

- a unary relation $\alpha \rightarrow \alpha' \in [\mathsf{d}(\_)]$ for each $\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$.

We write $\langle t, \alpha\rangle \not\mapsto$ for the set of all transition formulas $\neg(\langle t, \alpha\rangle \xmapsto{r,\rho} \langle t', \alpha'\rangle)$ where t' is a closed term of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$, $\alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$, $r \in \mathbb{R}^>$ and $\rho \in \mathcal{E}_r$.

The auxiliary *discontinuity* relations $\alpha \rightarrow \alpha' \in [\mathsf{d}(\_)]$ can be explained as follows:

- $\alpha \rightarrow \alpha' \in [\mathsf{d}(t)]$: in state α, the discontinuities resulting from a transition to state α' are possible for process t.

The following is an important property of the transition relations defined by the transition rules given for $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. We have for all closed terms t, for all $\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$:

$$\alpha \rightarrow \alpha' \in [\mathsf{d}(t)] \text{ implies } \alpha \in [\mathsf{s}(t)] \,.$$

Table 16
Additional rules for $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ $(a, b, c \in \mathsf{A},\ r > 0)$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle,\ \alpha \rightarrow \alpha' \in [\mathsf{d}(y)],\ \alpha' \in [\mathsf{s}(y)]}{\langle x \parallel y,\alpha\rangle \xrightarrow{a} \langle x' \parallel y,\alpha'\rangle} \qquad \frac{\alpha \rightarrow \alpha' \in [\mathsf{d}(x)],\ \alpha' \in [\mathsf{s}(x)],\ \langle y,\alpha\rangle \xrightarrow{a} \langle y',\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xrightarrow{a} \langle x \parallel y',\alpha'\rangle}$$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle,\ \alpha \rightarrow \alpha' \in [\mathsf{d}(y)],\ \alpha' \in [\mathsf{s}(y)]}{\langle x \parallel y,\alpha\rangle \xrightarrow{a} \langle y,\alpha'\rangle} \qquad \frac{\alpha \rightarrow \alpha' \in [\mathsf{d}(x)],\ \alpha' \in [\mathsf{s}(x)],\ \langle y,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xrightarrow{a} \langle x,\alpha'\rangle}$$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle y',\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xrightarrow{c} \langle x' \parallel y',\alpha'\rangle}\ \gamma(a,b) = c \qquad \frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle \surd,\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xrightarrow{c} \langle x',\alpha'\rangle}\ \gamma(a,b) = c$$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle y',\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xrightarrow{c} \langle y',\alpha'\rangle}\ \gamma(a,b) = c \qquad \frac{\langle x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle \surd,\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xrightarrow{c} \langle \surd,\alpha'\rangle}\ \gamma(a,b) = c$$

$$\frac{\langle x,\alpha\rangle \xmapsto{r,\rho} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xmapsto{r,\rho} \langle y',\alpha'\rangle}{\langle x \parallel y,\alpha\rangle \xmapsto{r,\rho} \langle x' \parallel y',\alpha'\rangle}$$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle,\ \alpha \rightarrow \alpha' \in [\mathsf{d}(y)],\ \alpha' \in [\mathsf{s}(y)]}{\langle x \between y,\alpha\rangle \xrightarrow{a} \langle x' \parallel y,\alpha'\rangle} \qquad \frac{\langle x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle,\ \alpha \rightarrow \alpha' \in [\mathsf{d}(y)],\ \alpha' \in [\mathsf{s}(y)]}{\langle x \between y,\alpha\rangle \xrightarrow{a} \langle y,\alpha'\rangle}$$

$$\frac{\langle x,\alpha\rangle \xmapsto{r,\rho} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xmapsto{r,\rho} \langle y',\alpha'\rangle}{\langle x \between y,\alpha\rangle \xmapsto{r,\rho} \langle x' \between y',\alpha'\rangle}$$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle y',\alpha'\rangle}{\langle x \mid y,\alpha\rangle \xrightarrow{c} \langle x' \parallel y',\alpha'\rangle}\ \gamma(a,b) = c \qquad \frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle \surd,\alpha'\rangle}{\langle x \mid y,\alpha\rangle \xrightarrow{c} \langle x',\alpha'\rangle}\ \gamma(a,b) = c$$

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle y',\alpha'\rangle}{\langle x \mid y,\alpha\rangle \xrightarrow{c} \langle y',\alpha'\rangle}\ \gamma(a,b) = c \qquad \frac{\langle x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle,\ \langle y,\alpha\rangle \xrightarrow{b} \langle \surd,\alpha'\rangle}{\langle x \mid y,\alpha\rangle \xrightarrow{c} \langle \surd,\alpha'\rangle}\ \gamma(a,b) = c$$

$$\frac{\langle x,\alpha\rangle \xmapsto{r,\rho} \langle x',\alpha'\rangle,\ \langle y,\alpha\rangle \xmapsto{r,\rho} \langle y',\alpha'\rangle}{\langle x \mid y,\alpha\rangle \xmapsto{r,\rho} \langle x' \mid y',\alpha'\rangle}$$

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

$$\frac{\langle x,\alpha\rangle \xmapsto{r,\rho} \langle x',\alpha'\rangle}{\langle \partial_H(x),\alpha\rangle \xmapsto{r,\rho} \langle \partial_H(x'),\alpha'\rangle}$$

$$\frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)]}{\alpha \in [\mathsf{s}(x \parallel y)]} \qquad \frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)]}{\alpha \in [\mathsf{s}(x \between y)]} \qquad \frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)]}{\alpha \in [\mathsf{s}(x \mid y)]} \qquad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(\partial_H(x))]}$$

Note that we have for all closed terms t in which the signal evolution operator only occurs in subterms of the form $\phi \curvearrowright_{\emptyset} t'$, for all $\alpha, \alpha' : \mathsf{V} \cup \dot{\mathsf{V}} \to \mathbb{R}$:

$$\alpha \rightarrow \alpha' \in [\mathsf{d}(t)] \text{ iff } \alpha \in [\mathsf{s}(t)]\ .$$

Hence, the auxiliary transition relations $\alpha \rightarrow \alpha' \in [\mathsf{d}(\_)]$ would be superfluous if the preclusion of discontinuities for certain state variables in state evolutions was not supported. We also have for all closed terms t and states α and α':

$$\alpha \rightarrow \alpha' \in [\mathsf{d}(t)] \text{ iff } \alpha \rightarrow \alpha' \models \mathsf{d}_\rho(t)\ .$$

Table 17
Rules for $\alpha \to \alpha' \in [\mathsf{d}(\_)]$ ($a \in \mathsf{A}_\delta$, $r > 0$)

$$\frac{}{\alpha \to \alpha' \in [\mathsf{d}(\tilde{a})]} \quad \frac{\alpha \to \alpha' \in [\mathsf{d}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(\sigma^0_{\mathsf{rel}}(x))]} \quad \frac{}{\alpha \to \alpha' \in [\mathsf{d}(\sigma^r_{\mathsf{rel}}(x))]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \alpha \to \alpha' \in [\mathsf{d}(y)]}{\alpha \to \alpha' \in [\mathsf{d}(x + y)]} \quad \frac{\alpha \to \alpha' \in [\mathsf{d}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(x \cdot y)]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(\psi :\rightharpoonup x)]} \quad \frac{}{\alpha \to \alpha' \in [\mathsf{d}(\psi :\rightharpoonup x)]}\ \alpha \not\models \psi \quad \frac{\alpha \to \alpha' \in [\mathsf{d}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(\psi \curvearrowright x)]}\ \alpha \models \psi$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \langle x, \alpha\rangle \xrightarrow{r,\rho} \langle x', \alpha''\rangle}{\alpha \to \alpha' \in [\mathsf{d}(\phi \,\triangledown_V\, x)]}\ \alpha \to \alpha' \models \mathcal{C}_V,\ \alpha \models \phi \quad \frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \langle x, \alpha\rangle \not\to}{\alpha \to \alpha' \in [\mathsf{d}(\phi \,\triangledown_V\, x)]}\ \alpha \models \phi$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(\chi \,\triangledown\, x)]} \quad \frac{}{\alpha \to \alpha' \in [\mathsf{d}(\chi \,\triangledown\, x)]}\ \alpha \not\models {}^\circ\chi \quad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(\nu_{\mathsf{rel}}(x))]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \alpha \to \alpha' \in [\mathsf{d}(y)],\ \langle x \parallel y, \alpha\rangle \xrightarrow{r,\rho} \langle x', \alpha''\rangle}{\alpha \to \alpha' \in [\mathsf{d}(x \parallel y)]} \quad \frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)],\ \langle x \parallel y, \alpha\rangle \not\to}{\alpha \to \alpha' \in [\mathsf{d}(x \parallel y)]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \alpha \to \alpha' \in [\mathsf{d}(y)],\ \langle x \between y, \alpha\rangle \xrightarrow{r,\rho} \langle x', \alpha''\rangle}{\alpha \to \alpha' \in [\mathsf{d}(x \between y)]} \quad \frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)],\ \langle x \between y, \alpha\rangle \not\to}{\alpha \to \alpha' \in [\mathsf{d}(x \between y)]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \alpha \to \alpha' \in [\mathsf{d}(y)],\ \langle x \mid y, \alpha\rangle \xrightarrow{r,\rho} \langle x', \alpha''\rangle}{\alpha \to \alpha' \in [\mathsf{d}(x \mid y)]} \quad \frac{\alpha \in [\mathsf{s}(x)],\ \alpha \in [\mathsf{s}(y)],\ \langle x \mid y, \alpha\rangle \not\to}{\alpha \to \alpha' \in [\mathsf{d}(x \mid y)]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)]}{\alpha \to \alpha' \in [\mathsf{d}(\partial_H(x))]}$$

$$\frac{\{\alpha \to \alpha' \in [\mathsf{d}(F(q))] \mid q \in U\}}{\alpha \to \alpha' \in [\mathsf{d}(\int_{u \in U} F(u))]} \quad \frac{\alpha \to \alpha' \in [\mathsf{d}(\langle t_X | E\rangle)]}{\alpha \to \alpha' \in [\mathsf{d}(\langle X | E\rangle)]}\ X = t_X \in E$$

4.5 Bisimulation and Soundness

The definitions of bisimulation equivalence and ic-bisimulation equivalence have to be adapted to the addition of discontinuity relations. The following condition must be added to both definitions:

- whenever $\alpha \to \alpha' \in [\mathsf{d}(t_1)]$, then $\alpha \to \alpha' \in [\mathsf{d}(t_2)]$.

The following example shows that bisimulation equivalence is not preserved by all operators of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$. We have

$$(v^\bullet = 1) \,\triangledown\, \tilde{\tilde{a}} \cdot ((v = 0) :\rightharpoonup \tilde{\tilde{b}}) \leftrightarrows (v^\bullet = 1) \,\triangledown\, \tilde{\tilde{a}} \cdot \tilde{\tilde{\delta}}\ .$$

First, we take the left-hand side as the first argument of a left merge with $(v^\bullet = 0) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{c}}$. A possible sequence of transitions is

$$\begin{aligned}
&\langle((v^\bullet = 1) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{a}} \cdot ((v = 0) :\to \tilde{\tilde{b}})) \mathbin{\lfloor\!\lfloor} ((v^\bullet = 0) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{c}}), v \mapsto *\rangle \\
&\quad \xrightarrow{a} \langle((v = 0) :\to \tilde{\tilde{b}}) \parallel ((v^\bullet = 0) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{c}}), v \mapsto 1\rangle \\
&\quad \xrightarrow{c} \langle(v = 0) :\to \tilde{\tilde{b}}, v \mapsto 0\rangle \\
&\quad \xrightarrow{b} \langle\surd, v \mapsto 0\rangle \;.
\end{aligned}$$

Here, $v \mapsto r$ denotes the state in which the value of v is r; and $*$ is any real number. Secondly, we take the right-hand side as the first argument of a left merge with $(v^\bullet = 0) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{c}}$. The only possible sequence of transitions starting from the same state is

$$\begin{aligned}
&\langle((v^\bullet = 1) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{a}} \cdot \tilde{\tilde{\delta}}) \mathbin{\lfloor\!\lfloor} ((v^\bullet = 0) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{c}}), v \mapsto *\rangle \\
&\quad \xrightarrow{a} \langle\tilde{\tilde{\delta}} \parallel ((v^\bullet = 0) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{c}}), v \mapsto 1\rangle \\
&\quad \xrightarrow{c} \langle\tilde{\tilde{\delta}}, v \mapsto 0\rangle \;.
\end{aligned}$$

This discrepancy does not occur with ic-bisimulation equivalence because

$$(v^\bullet = 1) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{a}} \cdot ((v = 0) :\to \tilde{\tilde{b}}) \mathrel{\not\leftrightarrows} (v^\bullet = 1) \mathbin{\ulcorner\!\!\!\cdot} \tilde{\tilde{a}} \cdot \tilde{\tilde{\delta}} \;.$$

Ic-bisimulation equivalence is preserved by all operators of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$.

Theorem 11 (Congruence) *Ic-bisimulation equivalence is a congruence with respect to the operators of* $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$.

PROOF. For ic-bisimulation equivalence, congruence follows immediately from the following. The transition rules for $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ constitute a complete transition system specification in panth format, and ic-bisimulation equivalence is the equivalence which is guaranteed to be a congruence in that case (see e.g. Refs. [40,36]).[5] $\square$

The axioms and lifting rule of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ are sound with respect to ic-bisimulation equivalence.

Theorem 12 (Soundness) *For all closed terms* t_1 *and* t_2 *of* $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$*, we have* $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}} \vdash t_1 = t_2$ *implies* $t_1 \leftrightarrows t_2$.

[5] This equivalence is called bisimulation equivalence in Refs. [40,36]. This should not be confused with what is called bisimulation equivalence in this paper.

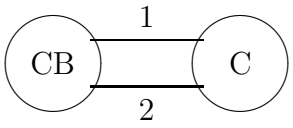

Fig. 1. Connection diagram for bottle filling system

PROOF. See Appendix A.5. □

As a corollary of Theorems 7 and 12, we have the soundness of two-phase derivation.

Corollary 13 (Soundness) *For all closed terms* t_1 *and* t_2 *of* $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$*, we have* $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}/\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}} \vdash_2 t_1 = t_2$ *implies* $t_1 \leftrightarrows t_2$*.*

4.6 Example: Bottle Filling System

In this section, we consider a bottle filling system. This example is adapted from Ref. [37]. The bottle filling system consists of two subsystems, a conveyer belt *CB* and a container *C*, which proceed concurrently. They communicate with each other at ports 1 and 2. The configuration of the bottle filling system is shown in Figure 1. We take the following informal description of the bottle filling system as the starting point of our specifications of the conveyer belt and the container.

Bottles on a conveyer belt are filled with 10 L of liquid poured from a container. When a bottle is put under the container, a tap is opened and the bottle is filled at a rate of 3 L/s until the container becomes empty or the bottle becomes full, whatever happens first. In the case where the container becomes empty first, the bottle is filled further at the same rate as the container. When the bottle is full, the tap is closed and the conveyer belt starts moving to put the next bottle under the container, which takes 1 s. The container is filled at a constant rate of r L/s. Its capacity is m L. Naturally, it is highly preferable that overflow never occurs. It is also preferable that the container does not get empty during the filling of each bottle. It is assumed that initially the conveyer belt starts moving to put the first bottle under the container and the container is half full.

The recursive specifications of the conveyer belt and the container given below need no further explanation because they are fairly direct representations of the corresponding informal descriptions. The recursive specification of the

conveyer belt consists of the following equations:

$$\begin{aligned}
CB &= (b = 0 \wedge \dot{b} = 0) \trianglerighteq \sigma^1_{\mathsf{rel}}\Big((b^\bullet = {}^\bullet b) \trianglerighteq \widetilde{s_1(start)} \cdot CB^{\mathrm{nf}}\Big)\,,\\
CB^{\mathrm{nf}} &= (b \leq 10 \wedge \dot{b} = 3) \trianglerighteq\\
&\qquad \Big(\sigma^*_{\mathsf{rel}}\Big((b^\bullet = {}^\bullet b) \trianglerighteq \widetilde{r_2(empty)} \cdot CB^{\mathrm{sf}}\Big)\\
&\qquad\quad + \sigma^*_{\mathsf{rel}}\Big((b = 10) :\rightarrow \Big((b^\bullet = 0) \trianglerighteq \widetilde{s_1(stop)} \cdot CB\Big)\Big)\Big)\,,\\
CB^{\mathrm{sf}} &= (b \leq 10 \wedge \dot{b} = r) \trianglerighteq\\
&\qquad \sigma^*_{\mathsf{rel}}\Big((b = 10) :\rightarrow \Big((b^\bullet = 0) \trianglerighteq \widetilde{s_1(stop)} \cdot CB\Big)\Big)\,.
\end{aligned}$$

The recursive specification of the container consists of the following equations:

$$\begin{aligned}
C &= (c = m/2) \trianglerighteq C^{\mathrm{inc}}\,,\\
C^{\mathrm{inc}} &= (c \leq m \wedge \dot{c} = r) \trianglerighteq\\
&\qquad \Big(\sigma^*_{\mathsf{rel}}\Big((c < m) :\rightarrow \Big((c^\bullet = {}^\bullet c) \trianglerighteq \widetilde{r_1(start)} \cdot C^{\mathrm{dec}}\Big)\Big)\\
&\qquad\quad + \sigma^*_{\mathsf{rel}}\Big((c = m) :\rightarrow \Big((c^\bullet = {}^\bullet c) \trianglerighteq \widetilde{overflow} \cdot \tilde{\delta}\Big)\Big)\Big)\,,\\
C^{\mathrm{dec}} &= (c \geq 0 \wedge \dot{c} = r - 3) \trianglerighteq\\
&\qquad \Big(\sigma^*_{\mathsf{rel}}\Big((c > 0) :\rightarrow \Big((c^\bullet = {}^\bullet c) \trianglerighteq \widetilde{r_1(stop)} \cdot C^{\mathrm{inc}}\Big)\Big)\\
&\qquad\quad + \sigma^*_{\mathsf{rel}}\Big((c = 0) :\rightarrow \Big((c^\bullet = {}^\bullet c) \trianglerighteq \widetilde{s_2(empty)} \cdot C^{\mathrm{dry}}\Big)\Big)\Big)\,,\\
C^{\mathrm{dry}} &= (c = 0) \trianglerighteq \sigma^*_{\mathsf{rel}}\Big((c^\bullet = {}^\bullet c) \trianglerighteq \widetilde{r_1(stop)} \cdot C^{\mathrm{inc}}\Big)\,.
\end{aligned}$$

The whole system is described by the following term:

$$\partial_H(CB \parallel C)\,,$$

where

$$\begin{aligned}
H = {} & \{s_1(d) \mid d \in \{start, stop\}\} \cup \{r_1(d) \mid d \in \{start, stop\}\}\\
& \cup \{s_2(empty)\} \cup \{r_2(empty)\}
\end{aligned}$$

and the communication function γ is defined such that

$$\gamma(s_i(d), r_i(d)) = \gamma(r_i(d), s_i(d)) = c_i(d)$$

for all $d \in \{start, stop, empty\}$ and $i \in \{1, 2\}$, and it is undefined otherwise.

Using the axioms and lifting rule of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$+INT+REC and the axioms and lifting rules of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$+INT+REC, we obtain by means of two-phase derivation the following guarded recursive specification of the whole system:

$$X^{\mathrm{ini}} = (c = m/2) \trianglerighteq X^{\mathrm{mv}}_{m/2} \,,$$

$$X^{\mathrm{mv}}_{c'} = (b = 0 \wedge \dot{b} = 0 \wedge c \leq m \wedge \dot{c} = r) \trianglerighteq \sigma^{1}_{\mathsf{rel}}\Big((b^{\bullet} = {}^{\bullet}b \wedge c^{\bullet} = {}^{\bullet}c) \trianglerighteq \widetilde{c_1(\mathit{start})} \cdot X^{\mathrm{nf}}_{c'+r}\Big)$$

(for every $c' < m - r$),

$$X^{\mathrm{mv}}_{c'} = (b = 0 \wedge \dot{b} = 0 \wedge c \leq m \wedge \dot{c} = r) \trianglerighteq \sigma^{(m-c')/r}_{\mathsf{rel}}\Big((b^{\bullet} = {}^{\bullet}b \wedge \dot{b}^{\bullet} = {}^{\bullet}\dot{b} \wedge c^{\bullet} = {}^{\bullet}c) \trianglerighteq \widetilde{\mathit{overflow}} \cdot \tilde{\tilde{\delta}}\Big)$$

(for every $m - r \leq c' < m$),

$$X^{\mathrm{nf}}_{c'} = (b \leq 10 \wedge \dot{b} = 3 \wedge c \geq 0 \wedge \dot{c} = r - 3) \trianglerighteq \sigma^{10/3}_{\mathsf{rel}}\Big((b^{\bullet} = 0 \wedge c^{\bullet} = {}^{\bullet}c) \trianglerighteq \widetilde{c_1(\mathit{stop})} \cdot X^{\mathrm{mv}}_{c'-(3-r)(10/3)}\Big)$$

(for every $(3 - r)(10/3) < c' < m$),

$$X^{\mathrm{nf}}_{c'} = (b \leq 10 \wedge \dot{b} = 3 \wedge c \geq 0 \wedge \dot{c} = r - 3) \trianglerighteq \sigma^{c'/(3-r)}_{\mathsf{rel}}\Big((b^{\bullet} = {}^{\bullet}b \wedge c^{\bullet} = {}^{\bullet}c) \trianglerighteq \widetilde{c_2(\mathit{empty})} \cdot X^{\mathrm{sf}}_{3c'/(3-r)}\Big)$$

(for every $c' \leq (3 - r)(10/3)$),

$$X^{\mathrm{sf}}_{b'} = (b \leq 10 \wedge \dot{b} = r \wedge c = 0) \trianglerighteq \sigma^{(10-b')/r}_{\mathsf{rel}}\Big((b^{\bullet} = 0 \wedge c^{\bullet} = {}^{\bullet}c) \trianglerighteq \widetilde{c_1(\mathit{stop})} \cdot X^{\mathrm{mv}}_{0}\Big)$$

(for every $b' \leq 10$).

From this recursive specification, it is easy to see that the contents c' of the container fluctuates around $m/2$ liters and overflow never occurs if $r = 30/13$ and $m/2 > r$. If $r > 30/13$, eventually overflow occurs. If $r < 30/13$, overflow never occurs but during the filling of each bottle the container gets empty.

4.7 Example: Railroad Crossing System

In this section, we consider a railroad crossing system. This example is adapted from Ref. [6]. The configuration of the railroad crossing system is shown in Figure 2. Analysis meant to provide answers to various basic questions about the railroad crossing system requires that the behaviour of its controller as well as the behaviour of the trains and the gate is described. We take the

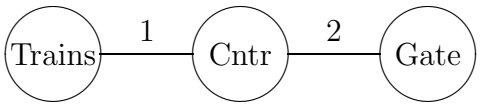

Fig. 2. Connection diagram for railroad crossing system

following informal description of the railroad crossing system from Ref. [37] as the starting point of our specifications.

When a train approaches the gate from a great distance its speed is between 48 m/s and 52 m/s. As soon as it passes the detector placed at 1000 m backward from the gate, an *appr* signal is sent to the controller. The train may now slow down, but its speed stays between 40 m/s and 52 m/s, and pass the gate. As soon as it passes the detector placed at 100 m forward from the gate, an *exit* signal is sent to the controller. A new train may come after the current one has passed the second detector, but only at a distance greater than or equal to 1500 m. The gate is able to receive *lower* and *raise* signals from the controller at any time. As soon as the gate receives a *lower* signal, it lowers from 90° to 0° at a constant rate of 20° per second. As soon as it receives a *raise* signal, it raises from 0° to 90° at the same rate. The controller is able to receive *appr* and *exit* signals from the train detectors at any time. When the controller receives an *appr* signal, it takes less than 5 s before a *lower* signal is sent to the gate. When the controller receives an *exit* signal, it takes less than 5 s before a *raise* signal is sent to the gate. Because of fault tolerance considerations, *appr* signals should always cause the gate to go down, and *exit* signals should be ignored while the gate is going down. It is assumed that initially there is no train at a distance smaller than 1400 m backward from the gate, the gate is open, and the controller is idling. Moreover, it is assumed that each single train changes its speed only smoothly.

It is worth mentioning that the identity of the trains passing the gate is not relevant to the analysis of the functioning of the railroad crossing system. Whatever the trains, the railroad crossing system treats them all the same.

The recursive specifications of the train movement, the gate and the controller given below need no further explanation because they are fairly direct representations of the corresponding informal descriptions. The recursive spec-

ification of the train movement consists of the following equations:

$$
\begin{aligned}
\mathit{Trains} &= (x \leq -1400) \curlywedge T^{\mathrm{far}} \,, \\
T^{\mathrm{far}} &= (x \leq -1000 \wedge 48 \leq \dot{x} \leq 52) \curlyvee \\
&\quad \sigma^{*}_{\mathsf{rel}}\Big((x = -1000) :\!\rightarrow \Big((x^{\bullet} = {}^{\bullet}x \wedge \dot{x}^{\bullet} = {}^{\bullet}\dot{x}) \curlyvee \widetilde{s_1(\mathit{appr})} \cdot T^{\mathrm{near}}\Big)\Big) \,, \\
T^{\mathrm{near}} &= (-1000 \leq x \leq 0 \wedge 40 \leq \dot{x} \leq 52) \curlyvee \\
&\quad \sigma^{*}_{\mathsf{rel}}\Big((x = 0) :\!\rightarrow \Big((x^{\bullet} = {}^{\bullet}x \wedge \dot{x}^{\bullet} = {}^{\bullet}\dot{x}) \curlyvee \widetilde{\mathit{pass}} \cdot T^{\mathrm{past}}\Big)\Big) \,, \\
T^{\mathrm{past}} &= (0 \leq x \leq 100 \wedge 40 \leq \dot{x} \leq 52) \curlyvee \\
&\quad \sigma^{*}_{\mathsf{rel}}\Big((x = 100) :\!\rightarrow \Big((x^{\bullet} \leq -1400) \curlyvee \widetilde{s_1(\mathit{exit})} \cdot T^{\mathrm{far}}\Big)\Big) \,.
\end{aligned}
$$

The recursive specification of the gate consists of the following equations:

$$
\begin{aligned}
\mathit{Gate} &= (r = 90) \curlywedge G^{\mathrm{op}} \,, \\
G^{\mathrm{op}} &= (r = 90 \wedge \dot{r} = 0) \curlyvee \\
&\quad \Big(\sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{lower})} \cdot G^{\mathrm{dn}}\Big) \\
&\qquad + \sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{raise})} \cdot G^{\mathrm{op}}\Big)\Big) \,, \\
G^{\mathrm{dn}} &= (0 \leq r \leq 90 \wedge \dot{r} = -20) \curlyvee \\
&\quad \Big(\sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{lower})} \cdot G^{\mathrm{dn}}\Big) \\
&\qquad + \sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{raise})} \cdot G^{\mathrm{up}}\Big) \\
&\qquad + \sigma^{*}_{\mathsf{rel}}\Big((r = 0) :\!\rightarrow \Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{\mathit{ready}_{\mathrm{dn}}} \cdot G^{\mathrm{cl}}\Big)\Big)\Big) \,, \\
G^{\mathrm{cl}} &= (r = 0 \wedge \dot{r} = 0) \curlyvee \\
&\quad \Big(\sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{lower})} \cdot G^{\mathrm{cl}}\Big) \\
&\qquad + \sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{raise})} \cdot G^{\mathrm{up}}\Big)\Big) \,, \\
G^{\mathrm{up}} &= (0 \leq r \leq 90 \wedge \dot{r} = 20) \curlyvee \\
&\quad \Big(\sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{lower})} \cdot G^{\mathrm{dn}}\Big) \\
&\qquad + \sigma^{*}_{\mathsf{rel}}\Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{r_2(\mathit{raise})} \cdot G^{\mathrm{up}}\Big) \\
&\qquad + \sigma^{*}_{\mathsf{rel}}\Big((r = 90) :\!\rightarrow \Big((r^{\bullet} = {}^{\bullet}r) \curlyvee \widetilde{\mathit{ready}_{\mathrm{up}}} \cdot G^{\mathrm{op}}\Big)\Big)\Big) \,.
\end{aligned}
$$

The recursive specification of the controller consists of the following equations:

$$Cntr = (d = 0) \trianglerighteq C^{\mathrm{idle}} ,$$

$$\begin{aligned} C^{\mathrm{idle}} &= (\dot{d} = 0) \trianglerighteq \\ &\quad \Big(\sigma^*_{\mathsf{rel}}\Big((d^\bullet = 0) \trianglerighteq \widetilde{r_1(appr)} \cdot C^{\mathrm{dn}}\Big) \\ &\qquad + \sigma^*_{\mathsf{rel}}\Big((d^\bullet = 0) \trianglerighteq \widetilde{r_1(exit)} \cdot C^{\mathrm{up}}\Big)\Big) , \\ C^{\mathrm{dn}} &= (0 \leq d \leq 5 \wedge \dot{d} = 1) \trianglerighteq \\ &\quad \Big(\sigma^*_{\mathsf{rel}}\Big((d^\bullet = 0) \trianglerighteq \widetilde{s_2(lower)} \cdot C^{\mathrm{idle}}\Big) \\ &\qquad + \sigma^*_{\mathsf{rel}}\Big((d^\bullet = {}^\bullet d) \trianglerighteq \widetilde{r_1(appr)} \cdot C^{\mathrm{dn}}\Big) \\ &\qquad + \sigma^*_{\mathsf{rel}}\Big((d^\bullet = {}^\bullet d) \trianglerighteq \widetilde{r_1(exit)} \cdot C^{\mathrm{dn}}\Big)\Big) , \\ C^{\mathrm{up}} &= (0 \leq d \leq 5 \wedge \dot{d} = 1) \trianglerighteq \\ &\quad \Big(\sigma^*_{\mathsf{rel}}\Big((d^\bullet = 0) \trianglerighteq \widetilde{s_2(raise)} \cdot C^{\mathrm{idle}}\Big) \\ &\qquad + \sigma^*_{\mathsf{rel}}\Big((d^\bullet = 0) \trianglerighteq \widetilde{r_1(appr)} \cdot C^{\mathrm{dn}}\Big) \\ &\qquad + \sigma^*_{\mathsf{rel}}\Big((d^\bullet = {}^\bullet d) \trianglerighteq \widetilde{r_1(exit)} \cdot C^{\mathrm{up}}\Big)\Big) . \end{aligned}$$

The whole system is described by the following term:

$$\partial_H(\mathit{Trains} \parallel \mathit{Cntr} \parallel \mathit{Gate}) ,$$

where

$$\begin{aligned} H = {} & \{s_1(d) \mid d \in \{\mathit{appr}, \mathit{exit}\}\} \cup \{r_1(d) \mid d \in \{\mathit{appr}, \mathit{exit}\}\} \\ & \cup \{s_2(d) \mid d \in \{\mathit{lower}, \mathit{raise}\}\} \cup \{r_2(d) \mid d \in \{\mathit{lower}, \mathit{raise}\}\} \end{aligned}$$

and the communication function γ is defined such that

$$\gamma(s_i(d), r_i(d)) = \gamma(r_i(d), s_i(d)) = c_i(d)$$

for all $d \in \{\mathit{appr}, \mathit{exit}, \mathit{lower}, \mathit{raise}\}$ and $i \in \{1, 2\}$, and it is undefined otherwise.

Using the axioms and lifting rule of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC and the axioms and lifting rules of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC, we obtain by means of two-phase derivation

the following guarded recursive specification of the whole system:

$$X^0 = (x \le -1400 \wedge d = 0 \wedge r = 90) \curlywedge X_0^1 \,,$$

$$X_{t'}^1 = (x \le -1000 \wedge 48 \le \dot{x} \le 52 \wedge \dot{d} = 0 \wedge r = 90 \wedge \dot{r} = 0) \curvearrowright \int_{t \in [400/52 - t', \infty)} \sigma_{\mathsf{rel}}^t\Big((d^\bullet = 0 \wedge \mathcal{C}_{\{x,r\}}) \curvearrowright c_1\widetilde{(appr)} \cdot X_0^2\Big)$$

(for every $t' < 90/20 + 5$),

$$X_{t'}^2 = (-1000 \le x \le 0 \wedge 40 \le \dot{x} \le 52 \wedge 0 \le d \le 5 \wedge \dot{d} = 1 \wedge r = 90 \wedge \dot{r} = 0) \curvearrowright \int_{t \in [0, 5 - t')} \sigma_{\mathsf{rel}}^t\Big((d^\bullet = 0 \wedge r^\bullet = {}^\bullet r \wedge \mathcal{C}_{\{x\}}) \curvearrowright c_2\widetilde{(lower)} \cdot X_{t'+t,90}^3\Big)$$

(for every $t' < 90/20 - (400/52 - 5)$),

$$X_{t',r}^3 = (-1000 \le x \le 0 \wedge 40 \le \dot{x} \le 52 \wedge \dot{d} = 0 \wedge 0 \le r \le 90 \wedge \dot{r} = -20) \curvearrowright \sigma_{\mathsf{rel}}^{r/20}\Big((r^\bullet = {}^\bullet r \wedge \mathcal{C}_{\{x,d\}}) \curvearrowright \widetilde{ready}_{\mathrm{dn}} \cdot X_{t'+r/20}^4\Big)$$

(for every $t' < 5$ and $r \le 90$),

$$X_{t'}^4 = (-1000 \le x \le 0 \wedge 40 \le \dot{x} \le 52 \wedge \dot{d} = 0 \wedge r = 0 \wedge \dot{r} = 0) \curvearrowright \int_{t \in [1000/52 - t', 1000/40 - t']} \sigma_{\mathsf{rel}}^t\Big(\mathcal{C}_{\{x,d,r\}} \curvearrowright \widetilde{pass} \cdot X^5\Big)$$

(for every $t' < 90/20 + 5$),

$$X^5 = (0 \le x \le 100 \wedge 40 \le \dot{x} \le 52 \wedge \dot{d} = 0 \wedge r = 0 \wedge \dot{r} = 0) \curvearrowright \int_{t \in [100/52, 100/40]} \sigma_{\mathsf{rel}}^t\Big((x^\bullet \le -1400 \wedge d^\bullet = 0 \wedge \mathcal{C}_{\{r\}}) \curvearrowright c_1\widetilde{(exit)} \cdot X^6\Big) \,,$$

$$X^6 = (x \le -1000 \wedge 48 \le \dot{x} \le 52 \wedge 0 \le d \le 5 \wedge \dot{d} = 1 \wedge r = 0 \wedge \dot{r} = 0) \curvearrowright \int_{t \in [0, 5)} \sigma_{\mathsf{rel}}^t\Big((d^\bullet = 0 \wedge r^\bullet = {}^\bullet r \wedge \mathcal{C}_{\{x\}}) \curvearrowright c_2\widetilde{(raise)} \cdot X_t^7\Big) \,,$$

$$X_{t'}^7 = (x \le -1000 \wedge 48 \le \dot{x} \le 52 \wedge \dot{d} = 0 \wedge 0 \le r \le 90 \wedge \dot{r} = 20) \curvearrowright \int_{t \in [400/52 - t', 90/20)} \sigma_{\mathsf{rel}}^t\Big((d^\bullet = 0 \wedge \mathcal{C}_{\{x,r\}}) \curvearrowright c_1\widetilde{(appr)} \cdot X_t^8\Big) + \sigma_{\mathsf{rel}}^{90/20}\Big((r^\bullet = {}^\bullet r \wedge \mathcal{C}_{\{x,d\}}) \curvearrowright \widetilde{ready}_{\mathrm{up}} \cdot X_{t'+90/20}^1\Big)$$

(for every $t' < 5$),

$$X_{t'}^8 = (-1000 \le x \le 0 \wedge 40 \le \dot{x} \le 52 \wedge 0 \le d \le 5 \wedge \dot{d} = 1 \wedge 0 \le r \le 90 \wedge \dot{r} = 20) \curvearrowright \int_{t \in [0, 90/20 - t')} \sigma_{\mathsf{rel}}^t\Big((d^\bullet = 0 \wedge r^\bullet = {}^\bullet r \wedge \mathcal{C}_{\{x\}}) \curvearrowright c_2\widetilde{(lower)} \cdot X_{t,20(t'+t)}^3\Big) + \sigma_{\mathsf{rel}}^{90/20 - t'}\Big((r^\bullet = {}^\bullet r \wedge \mathcal{C}_{\{x,d\}}) \curvearrowright \widetilde{ready}_{\mathrm{up}} \cdot X_{90/20 - t'}^2\Big)$$

(for every $400/52 - 5 < t' < 90/20$).

Recall that $\mathcal{C}_V$ abbreviates $\bigwedge_{v \in V}(v^\bullet = {}^\bullet v \wedge \dot{v}^\bullet = {}^\bullet \dot{v})$. From this recursive specification, it is not difficult to see that (1) a train can only pass the gate when the gate is closed, (2) the gate opens after a train has left the track unless a new train has entered the track and (3) the system reacts adequately when a new train enters the track while the gate is going up. Analysis of this recursive specification is sufficient for virtually all relevant safety and liveness properties of the system in this case where it is not the continuously changing state that has to be controlled. For example, although it is important to know when a train passes the gate, it is in this case not important to know where the train is during its approach. However, it is most likely different in those cases where it is the continuously changing state that has to be controlled.

In all cases, an important advantage of using the proposed process algebra for the description and analysis of hybrid systems is that one does not have to be finished with real analysis before one can use process algebra. For example, a process algebra with timing can only be used for the description and analysis of a hybrid system after all timing that arises from the continuous behaviour of the system has been determined with the help of real analysis – with the danger of abstracting too far – whereas real analysis is irrelevant in the stage where the process algebra with timing is used.

5 Localization

In this section, we extend $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ with localization. The localization operator makes it possible to keep discontinuities of a state variable local, in other words to inhibit discontinuities of the state variable caused by the environment. This extension can be useful in various applications. We illustrate this by means of an example concerning a vehicle with velocity control.

5.1 $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ *with Localization*

In order to support the preclusion of discontinuities for certain state variables due to actions performed by the environment of a process, we add localization to $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. Let P be a process and v be a state variable. Then the *localization* of P with respect to v, written $v \nabla P$, behaves like P, but with its state evolving without discontinuities for v whenever it is idling.

In the railroad crossing system described in Section 4.7, the signal evolution operator is consistently used in such a way that the states of *Trains*, *Gate* and *Cntr*, as well as consequently the state of the whole system, must always evolve during idling without discontinuities for all state variables. This is possible

Table 18
Axioms for localization ($a \in \mathsf{A}_\delta$, $r > 0$)

$v \nabla \bot = \bot$	HSL1
$v \nabla \tilde{\tilde{a}} = \tilde{\tilde{a}}$	HSL2
$v \nabla \sigma^r_{\mathsf{rel}}(x) = \mathsf{T} \mathbin{\curvearrowright}_{\{v\}} \sigma^r_{\mathsf{rel}}(v \nabla x)$	HSL3
$v \nabla (x + y) = v \nabla x + v \nabla y$	HSL4
$v \nabla x \cdot y = (v \nabla x) \cdot (v \nabla y)$	HSL5
$v \nabla (\psi :\rightarrow x) = \psi :\rightarrow (v \nabla x)$	HSL6
$v \nabla (\psi \mathbin{\curvearrowright\!\!\blacktriangle} x) = \psi \mathbin{\curvearrowright\!\!\blacktriangle} (v \nabla x)$	HSL7
$v \nabla (\phi \mathbin{\curvearrowright}_V x) = \phi \mathbin{\curvearrowright}_V (v \nabla x)$	HSL8
$v \nabla (\chi \mathbin{\curvearrowright} \tilde{\tilde{a}}) = \chi \mathbin{\curvearrowright} \tilde{\tilde{a}}$	HSL9
$v \nabla (v' \nabla x) = v' \nabla (v \nabla x)$	HSL10
$v \nabla (\int_{u \in U} F(u)) = \int_{u \in U} (v \nabla F(u))$	HSL11

because, for each state variable, there is only one process that may cause discontinuities of the state variable, there is only one process that behaves dependent on the value of the state variable, and those processes are the same. That is, the state variables x, r and d are local to the processes *Trains*, *Gate* and *Cntr*, respectively. With or without localization, these processes, as well as the whole system, behave exactly the same. In other systems, we sometimes find that some state variable is not local, but shared by two or more processes. This means that the signal evolution operator has to be used in such a way that the states of those processes may sometimes evolve during idling with discontinuities for that state variable. In such cases, localization of the whole system inhibits further discontinuities caused by its environment. It is worth noticing that real analysis would not be a great help to the analysis of the system, if its state could evolve with discontinuities when it is idling. The use of localization will be illustrated in Section 5.2.

The additional axioms for localization are the equations given in Table 18. Axioms HSL1–HSL11 show that localization is a global version of an instance of signal evolution.

We shall henceforth use the name $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC+HSL to refer to the extension of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC with localization, and likewise the name $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$+INT+REC+HSL.

The structural operational semantics for localization is described by the rules given in Table 19. Bisimulation equivalence and ic-bisimulation equivalence are preserved by localization. All additional axioms for localization are sound with respect to bisimulation equivalence and ic-bisimulation equivalence.

Table 19
Additional rules for localization ($a \in \mathsf{A}$, $r > 0$)

$$\frac{\langle x,\alpha\rangle \xrightarrow{a} \langle x',\alpha'\rangle}{\langle v \nabla x,\alpha\rangle \xrightarrow{a} \langle v \nabla x',\alpha'\rangle} \qquad \frac{\langle a,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle}{\langle v \nabla x,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle}$$

$$\frac{\langle x,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle x',\alpha'\rangle}{\langle v \nabla x,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle v \nabla x',\alpha'\rangle} \qquad \alpha \overset{r,\rho}{\longmapsto} \alpha' \models_{\{v\}} \mathsf{T} \qquad \frac{\alpha \in [\mathsf{s}(x)]}{\alpha \in [\mathsf{s}(v \nabla x)]}$$

$$\frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \langle x,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle x',\alpha''\rangle}{\alpha \to \alpha' \in [\mathsf{d}(v \nabla x)]} \qquad \alpha \to \alpha' \models \mathcal{C}_{\{v\}} \qquad \frac{\alpha \to \alpha' \in [\mathsf{d}(x)],\ \langle x,\alpha\rangle \not\mapsto}{\alpha \to \alpha' \in [\mathsf{d}(v \nabla x)]}$$

5.2 Example: Vehicle with Velocity Control

In this section, we consider a vehicle with velocity control. This example is adapted from Ref. [22]. The vehicle with velocity control consists of the vehicle and a controller. The vehicle follows a suggested acceleration a approximately, to within an error of ϵ. The velocity controller monitors the velocity v of the vehicle and produces a new suggested acceleration every d time units. The suggested acceleration is chosen in such a way that the velocity of the vehicle will remain below $v_{\max}$. We assume that the vehicle starts with velocity 0 and the velocity controller with suggested acceleration 0. We also assume that $v_{\max} \geq \epsilon\, d$. The recursive specification of the vehicle consists of the following equations:

$$V = (v = 0) \mathbin{\triangle} V' \,,$$

$$V' = (a - \epsilon \leq \dot{v} \leq a + \epsilon) \mathbin{\triangledown_{\{v\}}} \sigma^*_{\mathsf{rel}}(\tilde{\tilde{\delta}}) \,.$$

The recursive specification of the velocity controller consists of the following equations:

$$C = (a = 0) \mathbin{\triangle} C' \,,$$

$$C' = (\dot{a} = 0) \mathbin{\triangledown} \sigma^d_{\mathsf{rel}}\Big((^\bullet v + (a^\bullet + \epsilon)\, d \leq v_{\max}) \mathbin{\triangledown} \widetilde{\widetilde{suggest}} \cdot C'\Big) \,.$$

The vehicle with velocity control is described by the following term:

$$a \nabla (V \parallel C) \,.$$

The point is that the vehicle process V does not preclude discontinuities for a, which is updated every d time units by the controller process C. The localization operator is used to inhibit further discontinuities of a caused by the environment of the vehicle and its controller. In other words, only the controller can update the suggested acceleration of the vehicle, and in this way affect the velocity.

6 Concluding Remarks

A process algebra has been presented which makes it possible to deal with the behaviour of systems in which the instantaneous state transitions caused by performing actions are alternated with continuous state evolutions. It is intended as an algebraic framework for the description and analysis of hybrid systems. The inescapable interface of this framework with real analysis is isolated in special lifting rules to derive equations with the help of a mathematical theory that includes real analysis. The application of the framework has been illustrated by means of various examples. In the analysis of a thermostat, a bottle filling system and a railroad crossing system, the lifting rules turned out to be essential.

6.1 Discussion of Main Choices

The process algebra for hybrid systems proposed in this paper extends the process algebra with continuous relative timing from Ref. [11]. One of the reasons to extend a process algebra with timing is the following notable experience with the use of process algebra with timing for the description and analysis of hybrid systems (see e.g. Ref. [11]). In many cases, when all timing that arises from continuous behaviour is known, the details of continuous behaviour are not relevant to analysis of the system concerned with respect to all or virtually all properties expected from it, but the details of timing are still relevant.

The process algebra for hybrid systems proposed in this paper also extends the process algebra with propositional signals from Ref. [12]. The initial ideas about the use of a timed variant of the process algebra with propositional signals for hybrid systems were born while the second author was working on timed frames [41,42]. Similar ideas, born independently, were outlined in Ref. [43], but those ideas have never been worked out. To the best of our knowledge, the process algebra with propositional signals is the only process algebra that provides such a simple means as a proposition to represent the state of a process. When dealing with hybrid systems, a feature like that, extended to state transitions and state evolutions, is very common. Using the formalism of hybrid automata, for example, a hybrid system is described by means of initial, invariant, jump and flow conditions.

One of the reasons to build on existing theory is that it is considered to be good practice. We add only two operators to the combination of the process algebra with continuous relative timing from Ref. [11] and the process algebra with propositional signals from Ref. [12], viz. the signal evolution operator and the signal transition operator. The latter operator actually replaces the terminal

signal emission operator of the process algebra with propositional signals. The question arises whether the resulting process algebra contains superfluous operators by taking over the operators of two process algebras which have not been devised for hybrid systems. We do not need the conditional proceeding operator, because its effect can be mimicked by the signal transition operator, but its presence contributes to a clear comprehension of the whole. We need all other operators. They cannot be mimicked by each other and they are all indispensable in most descriptions of hybrid systems given in this paper.

The choice of the operators that have been added to the combination of the process algebra with continuous relative timing and the process algebra with propositional signals has been strongly influenced by the formalism of hybrid automata. As a consequence, there are close connections between the process algebra for hybrid systems and the formalism of hybrid automata. These connections are elaborated in ongoing work mentioned in Section 6.2.

Like in the formalism of hybrid automata, each switch from one continuous mode to another requires that an action is performed. This feature is clearly a consequence of our choice to build on the process algebra with continuous relative timing from Ref. [11]. It is directly inherited from that process algebra. Actions may not be needed to model switches between continuous modes of systems that behave purely according to physical laws, but we believe that it is seldom artificial to use them. Moreover, we believe that the feature discussed here is not really relevant to the degree of usefulness of the proposed process algebra. However, experience in practical applications is needed to make a firm claim.

6.2 Ongoing and Future Work

It was mentioned in the introduction that the process algebra proposed in this paper is inspired by the work on the formalism of hybrid automata. In ongoing work, we are elaborating the connections between the proposed process algebra and the formalism of hybrid automata. In Ref. [44], we show that hybrid automata can be faithfully represented using the proposed process algebra: the representations of two hybrid automata are bisimilar if and only if their standard interpretations as timed transition systems are bisimilar. [6] The representation of a hybrid automaton involves a recursive specification with

[6] The timed transition system associated with a hybrid automaton may have multiple initial states. We deal with that in the process algebra representation in the way described in Ref. [45].

an equation of the form

$$X_m = \phi_m \mathbin{\triangledown}_V \left(\sum_{s \in S_m} \chi_s \mathbin{\triangledown} \widetilde{\widetilde{a_s}} \cdot X_{m'_s} + \int_{u \in (0,\infty)} \sigma^u_{\mathsf{rel}}(X_m) \right)$$

for each control mode m of the hybrid automaton concerned. It is not difficult to establish that the proposed process algebra has more expressive power than the formalism of hybrid automata. An important point is that not even all recursive equations of the form

$$X_m = \phi_m \mathbin{\triangledown}_V \left(\sum_{s \in S_m} \chi_s \mathbin{\triangledown} \widetilde{\widetilde{a_s}} \cdot X_{m'_s} + \sum_{i \in I_m} \psi_i :\rightarrow \left(\phi_{m_i} \mathbin{\triangledown}_V \int_{u \in (0,\infty)} \sigma^u_{\mathsf{rel}}(X_m) \right) \right)$$

can be reduced to an equation of the form used to represent hybrid automata.

We mentioned in Section 3.3 that we would like to add a hiding operator $v\,\Delta$ for each state variable v, but that this extension would require a semantics that carries more detail than the structural operational semantics given in this paper. Working out the addition of those hiding operators is one of the options for future work.

The new process algebra for hybrid systems proposed in this paper represents a large amount of work. Therefore, it is not amazing that it induces a lot of other options for future work. We mention only a few options. Development of efficient proof techniques is important because there is no effective procedure for determining of an arbitrary equation of the proposed process algebra whether it is derivable. Investigation into restricted versions of the proposed process algebra that make an effective procedure possible is also interesting. In continuation of the current work concerning the connections with the formalism of hybrid automata, it is interesting to investigate the adaptation of model checking tools developed for hybrid automata to restricted versions of the proposed process algebra. Together with that a suitable temporal logic should be developed. Of course, it is very important that case-studies to assess the degree of usefulness in practical applications are carried out in conjunction with all the theoretical work mentioned above. If the design of hybrid systems can indeed be improved by the results of that work, it is worth turning it into an industrial method that can be used by both software engineers and control engineers when designing hybrid systems. We mention that the proposed process algebra for hybrid systems has not been designed with the objective to make easy transfer to practical control engineering possible. It appears that HyPA [19], which is discussed in Section 6.3, has been designed with that objective.

Quite another option for further work is in the area of tool support for soundness proofs for process algebras. The creation of the full soundness proofs for $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ is a very time-consuming, but for the greater part routine, affair in which mistakes are easily made. It gave us a shock to experience that our initial soundness proofs contained a few mistakes. The making of a readable electronic version of the proofs by hand takes up a great deal of time too. All this calls for a semi-automatic way for proving soundness of equational axioms, with respect to common versions of bisimulation equivalence, from transition rules. We think of a tool that can carry out routine work such as searching for applicable transition rules, producing their relevant instances, checking and recording the proof steps made, and making a readable version of the proof while it is created.

6.3 Related Work

Concerning related work, we mention the early work on hybrid CSP [15], the recent work on the ϕ-calculus [17], and the very recent work on HyPA [19]. In hybrid CSP and the ϕ-calculus, which are variants of timed CSP and the π-calculus, respectively, one can only deal with continuous behaviour in a limited way. The main limitation of hybrid CSP, which dates back to 1994, is that parallel composition of processes is only possible if the continuous behaviour exhibited by the parallel processes is independent: the processes are not allowed to have a state variable in common. The main limitation of the ϕ-calculus is that the expressions constructed by means of the operators of the ϕ-calculus denote processes that do not exhibit continuous behaviour: continuous behaviour can only be exhibited by a special process, called an environment, which is described separately. The work on HyPA is the most closest to our work, and deserves a more detailed discussion.

HyPA is an extension of ACP for hybrid systems on which the first report appeared very shortly after the report version of this paper. In that report, it is stated that HyPA and $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ are very similar. We agree only in part. The transition systems induced by the structural operational semantics of HyPA are in some respects similar to the ones induced by the structural operational semantics of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. However, in our opinion, the similarities end with that. Here, we confine ourselves to mentioning some of the most important dissimilarities.

The operators of ACP are the only operators that HyPA and $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ have in common. The additional operators, which make it possible to deal with the behaviour of hybrid systems, are quite different. This dissimilarity has far-reaching consequences. The absence of operators for timing means that mere timing must be modelled in HyPA by means of state variables that behave as

clocks. Because $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ includes the operators of $\mathrm{ACP}^{\mathrm{srt}}$, we can transform the description of a hybrid system in $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ into one that makes explicit timing that arises from the evolution of its state. Similar transformations are not possible with descriptions of hybrid systems in HyPA.

Another important dissimilarity concerns alternative composition. The structural operational semantics of alternative composition in $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ provides for a form of time-determinism: if $\langle t, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t', \alpha'\rangle$ and $\langle t, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t'', \alpha'\rangle$, then $t' \equiv t''$. This property can be paraphrased roughly as follows: a choice between different idling processes is postponed so long as all can idle. However, a choice between different evolutions of the state is not postponed. The structural operational semantics of alternative composition in HyPA does not provide for a form of time-determinism. We consider the above-mentioned form of time-determinism of vital importance for a faithful representation of all time-dependent behaviour. In the case of behaviour of hybrid systems, the continuous state changes that take place during idling may surely make choices between processes available at certain points of time, but that does not amount to the property that such choices may happen during idling. It appears that this observation contradicts the main argument used in Ref. [19] against the form of time-determinism present in $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. The representation of hybrid automata in $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ sketched in Section 6.2 draws attention to the fact that, different from what is said in Ref. [19], this form of time-determinism is in line with the approach of hybrid automata. Because each control mode has just one alternative to proceed with idling, time-determinism is just not an issue.

Finally, we mention the loosely related work on duration calculus. The original duration calculus, called DC, is proposed in Ref. [46]. DC is an interval temporal logic designed for expressing and reasoning about assumptions and requirements on how the state of a real-time system changes over time. An extension of DC for hybrid systems, called EDC, is proposed in Ref. [47]. EDC can be used during requirement capturing and early design stages of the development of a hybrid systems. As soon as during the design details about actions taking place become relevant, a process algebra such as $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ is better suited. To investigate how the switch from EDC to $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ can be made in a semantically sound way is still another option for further work.

6.4 Miscellaneous Remarks

The process algebra for hybrid systems proposed in this paper does not incorporate abstraction from internal actions. This issue is not even fully understood in process algebras with timing. The version of branching bisimulation equivalence for processes with discrete relative timing proposed in Ref. [48] for

this purpose, and adapted to continuous relative timing in Ref. [11], is too fine for many applications. A slightly coarser equivalence is proposed in Ref. [49].

The proposed process algebra for hybrid systems does not exclude the possibility of two or more actions to be performed consecutively at the same point in time. For hybrid automata, this possibility is sometimes excluded. A variant of the proposed process algebra that excludes this possibility as well can be devised along similar lines as the process algebra with nonstandard timing from Ref. [25].

Concerning Zeno behaviour, the phenomenon that infinitely many instantaneous state changes happen before a certain point of time, the following remark is in order. The axioms and lifting rules given in this paper are based on a notion of bisimulation that does not distinguish between behaviours that occur after a point of time at which infinitely many instantaneous state changes accumulate. A notion of bisimulation to deal with Zeno behaviour is proposed in Ref. [50]. In our opinion, however, Zeno behaviour is primarily a sign that a questionable abstraction of a real system has been made; and behaviour occurring after such unrealizable behaviour is absolutely irrelevant.

The process algebra with continuous relative timing from Ref. [11] on which we build the proposed process algebra for hybrid systems arises from an attempt to streamline a lot of work on process algebra with timing in the setting of ACP done since 1989. It originates from $\mathrm{ACP}_{\mathrm{st}}$, a version of ACP with continuous relative timing from Ref. [51], which, unlike the earlier version of ACP with continuous relative timing from Ref. [52], does not combine performing an action with idling for a period of time. An interesting extension of the version of ACP with continuous absolute timing from Ref. [52] is the real space process algebra proposed in Ref. [53]. Still another option for further work is to investigate to what extent the examples concerning data transmission via a mobile intermediate station from Refs. [53,54] can be described using the process algebra for hybrid systems proposed in this paper.

Acknowledgements

The work presented in this paper has been partly carried out while the second author was also part-time at Utrecht University, Department of Philosophy. We thank Pieter Cuijpers from Eindhoven University of Technology, Computing Science Department, for pointing us at the existence of two serious errors in our initial soundness proofs. We have used variants of one of his counterexamples of soundness in Sections 4.3 and 4.5. We also thank the referees for their valuable comments.

Appendix

In this appendix, we outline elimination, congruence and soundness proofs for $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$. The full proofs are for the greater part very long and really tedious. They are, for example, much longer than the full proofs given in Ref. [55] for a version of ACP with discrete relative timing. We focus on the most difficult parts of the proofs in this appendix. Even for those parts, we do not give full details. That is, we mention the axioms by which the equations relevant to the elimination proofs can be derived instead of presenting the derivations of the equation, and we present in the congruence and soundness proofs the conditions under which transition relations hold without mentioning how the conditions follow from the transition rules. What is left out, can easily be found by consulting the axioms referred to or the applicable transition rules.

A.1 Proof of Theorem 1 (Elimination for $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$*)*

The proof is straightforward by induction on the structure of closed term t. For terms t of the forms $\bot$, $\tilde{\tilde{a}}$, $\sigma^r_{\mathsf{rel}}(t')$, $t' + t''$ and $\psi \between t'$, it is trivial to show that there is a basic term that is derivably equal to t. For terms t of the forms $\nu_{\mathsf{rel}}(t')$, $t' \cdot t''$, $\psi :\!\!\rightarrow t'$, $\phi \between_V t'$ and $\chi \between t'$, it follows immediately from the induction hypothesis and the following lemmas:

(1) for all $t \in \mathcal{B}$, there is a $t' \in \mathcal{B}$ such that $\nu_{\mathsf{rel}}(t) = t'$ is derivable;
(2) for all $t, t' \in \mathcal{B}$, there is a $t'' \in \mathcal{B}$ such that $t \cdot t' = t''$ is derivable;
(3) for all $\psi \in \mathcal{P}_{\mathsf{st}}$ and $t \in \mathcal{B}$, there is a $t' \in \mathcal{B}$ such that $\psi :\!\!\rightarrow t = t'$ is derivable;
(4) for all $\phi \in \mathcal{P}_{\mathsf{st}}$, $V \subseteq \mathsf{V}$ and $t \in \mathcal{B}$, there is a $t' \in \mathcal{B}$ such that $\phi \between_V t = t'$ is derivable;
(5) for all $\chi \in \mathcal{P}_{\mathsf{tr}}$ and $t \in \mathcal{B}$, there is a $t' \in \mathcal{B}$ such that $\chi \between t = t'$ is derivable.

These lemmas are easily proven by induction on the structure of basic term t. We present here the proof of the fourth lemma. The proofs of the other lemmas are similar, but less complicated.

The proof of the fourth lemma goes as follows:

- $\boldsymbol{t \equiv \bot}$: Then $\phi \between_V \bot = \bot$ by SE2, HSE10; and $\bot \in \mathcal{B}$.
- $\boldsymbol{t \equiv \psi \between \tilde{\tilde{\delta}}}$: Then $\phi \between_V (\psi \between \tilde{\tilde{\delta}}) = (\phi \wedge \psi) \between \tilde{\tilde{\delta}}$ by HSE10, HSE3, SE5. We proceed by distinguishing two cases:
  - $\phi \wedge \psi \in {\mathcal{P}_{\mathsf{st}}}^+$: Then $(\phi \wedge \psi) \between \tilde{\tilde{\delta}} \in \mathcal{B}$.
  - $\phi \wedge \psi \notin {\mathcal{P}_{\mathsf{st}}}^+$: Then $(\phi \wedge \psi) \between \tilde{\tilde{\delta}} = \bot$ by SE2; and $\bot \in \mathcal{B}$.
- $\boldsymbol{t \equiv \psi :\!\!\rightarrow (\chi \between \tilde{\tilde{a}})}$: Then $\phi \between_V (\psi :\!\!\rightarrow (\chi \between \tilde{\tilde{a}})) = \phi \between \tilde{\tilde{\delta}} + \psi :\!\!\rightarrow (\chi \between \tilde{\tilde{a}})$ by HSE9, HSE12, SE7, SE5, A6SR, SE3. We proceed by distinguishing two cases:
  - $\phi \in {\mathcal{P}_{\mathsf{st}}}^+$: Then $\phi \between \tilde{\tilde{\delta}} + \psi :\!\!\rightarrow (\chi \between \tilde{\tilde{a}}) \in \mathcal{B}$.

  - $\phi \notin {\mathcal{P}_{\mathsf{st}}}^+$: Then $\phi \between \tilde{\tilde{\delta}} + \psi :\rightarrow (\chi \between \tilde{\tilde{a}}) = \bot$ by SE3, SE2; and $\bot \in \mathcal{B}$.
- $\boldsymbol{t \equiv \psi :\rightarrow (\chi \between \tilde{\tilde{a}} \cdot t'')}$: Analogous to the previous case.
- $\boldsymbol{t \equiv \psi :\rightarrow (\phi' \between_{V'} \sigma^r_{\mathsf{rel}}(t''))}$: Then $\phi \between_V (\psi :\rightarrow (\phi' \between_{V'} \sigma^r_{\mathsf{rel}}(t''))) = \phi \between \tilde{\tilde{\delta}} + \psi :\rightarrow ((\phi \wedge \phi') \between_{V \cup V'} \sigma^r_{\mathsf{rel}}(t''))$ by HSE9, HSE11, A6SR, SE3. We proceed by distinguishing three cases:
  - $\phi \in {\mathcal{P}_{\mathsf{st}}}^+$ and $\phi \wedge \phi' \in {\mathcal{P}_{\mathsf{st}}}^+$: Then $\phi \between \tilde{\tilde{\delta}} + \psi :\rightarrow ((\phi \wedge \phi') \between_{V \cup V'} \sigma^r_{\mathsf{rel}}(t'')) \in \mathcal{B}$.
  - $\phi \in {\mathcal{P}_{\mathsf{st}}}^+$ and $\phi \wedge \phi' \notin {\mathcal{P}_{\mathsf{st}}}^+$: Then $\phi \between \tilde{\tilde{\delta}} + \psi :\rightarrow ((\phi \wedge \phi') \between_{V \cup V'} \sigma^r_{\mathsf{rel}}(t'')) = \bot$ by HSE2, SE2, SE7, GC3SR, SE3, SE5, A1, A6SR; and $\bot \in \mathcal{B}$.
  - $\phi \notin {\mathcal{P}_{\mathsf{st}}}^+$: Then $\phi \between \tilde{\tilde{\delta}} + \psi :\rightarrow ((\phi \wedge \phi') \between_{V \cup V'} \sigma^r_{\mathsf{rel}}(t'')) = \bot$ by SE2, NE1; and $\bot \in \mathcal{B}$.
- $\boldsymbol{t \equiv t'' + t'''}$: Then $\phi \between_V (t'' + t''') = \phi \between_V t'' + \phi \between_V t'''$ by HSE7. By the induction hypothesis there are basic terms t^* and t^{**} such that $\phi \between_V t'' = t^*$ and $\phi \between_V t''' = t^{**}$; and $t^* + t^{**} \in \mathcal{B}$.

A.2 Proof of Theorem 6 (Congruence for $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$*)*

For ic-bisimulation equivalence, congruence follows immediately from the following. The transition rules for $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ constitute a complete transition system specification in panth format, and ic-bisimulation equivalence is the equivalence which is guaranteed to be a congruence in that case (see e.g. Refs. [40,36]). [7]

For bisimulation equivalence, we prove for each operator of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ that it preserves bisimulation equivalence. We present here the proof for sequential composition. The proofs for the other operators of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ are similar. The proof for alternative composition is equally complicated, and the proofs for the remaining operators are less complicated.

Suppose that $t_1 \leftrightarrows t'_1$ and $t_2 \leftrightarrows t'_2$. For each state α, let R^1_α and R^2_α be bisimulation relations witnessing $\langle t_1, \alpha\rangle \leftrightarrows \langle t'_1, \alpha\rangle$ and $\langle t_2, \alpha\rangle \leftrightarrows \langle t'_2, \alpha\rangle$, respectively. We write R^i $(i = 1, 2)$ for the union of R^i_α over all states α. Let α_0 be a fixed but arbitrary state. Define

$$R_{\alpha_0} = R'_{\alpha_0} \cup R^2 \ ,$$

where

$$R'_{\alpha_0} = \{(\langle s_1 \cdot t_2, \alpha\rangle, \langle s'_1 \cdot t'_2, \alpha\rangle) \mid R^1_{\alpha_0}(\langle s_1, \alpha\rangle, \langle s'_1, \alpha\rangle)\} \ .$$

We show that R_{α_0} is a bisimulation relation. Suppose that $R_{\alpha_0}(\langle t, \alpha\rangle, \langle t', \alpha\rangle)$. In the case where $R^2(\langle t, \alpha\rangle, \langle t', \alpha\rangle)$, the conditions for a bisimulation relation

[7] This equivalence is called bisimulation equivalence in Refs. [40,36]. This should not be confused with what is called bisimulation equivalence in this paper.

are trivially satisfied; and in the case where $R'_{\alpha_0}(\langle t,\alpha\rangle,\langle t',\alpha\rangle)$, we may assume that $t \equiv s_1 \cdot t_2$, $t' \equiv s'_1 \cdot t'_2$ and $R^1_{\alpha_0}(\langle s_1,\alpha\rangle,\langle s'_1,\alpha\rangle)$. In the latter case, we distinguish between the different kinds of transition relations:

- *action step relations*: Suppose $\langle t,\alpha\rangle \xrightarrow{a} \langle u,\alpha'\rangle$. We proceed by distinguishing the two possibilities for u:
  - $u \equiv v \cdot t_2$: $\langle t,\alpha\rangle \xrightarrow{a} \langle u,\alpha'\rangle$ holds only if $\langle s_1,\alpha\rangle \xrightarrow{a} \langle v,\alpha'\rangle$. Because $R^1_{\alpha_0}(\langle s_1,\alpha\rangle,\langle s'_1,\alpha\rangle)$, there exists a v' such that $\langle s'_1,\alpha\rangle \xrightarrow{a} \langle v',\alpha'\rangle$ and $R^1_{\alpha_0}(\langle v,\alpha'\rangle,\langle v',\alpha'\rangle)$. So $\langle s'_1 \cdot t'_2,\alpha\rangle \xrightarrow{a} \langle v' \cdot t'_2,\alpha'\rangle$ and $R_{\alpha_0}(\langle v \cdot t_2,\alpha'\rangle,\langle v' \cdot t'_2,\alpha'\rangle)$.
  - $u \equiv t_2$: $\langle t,\alpha\rangle \xrightarrow{a} \langle u,\alpha'\rangle$ holds only if $\langle s_1,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle$ and $\alpha' \in [\mathsf{s}(t_2)]$. Because $R^1_{\alpha_0}(\langle s_1,\alpha\rangle,\langle s'_1,\alpha\rangle)$, we have $\langle s'_1,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle$. Moreover, because $R^2(\langle t_2,\alpha'\rangle,\langle t'_2,\alpha'\rangle)$, we have $\alpha' \in [\mathsf{s}(t'_2)]$. So $\langle s'_1 \cdot t'_2,\alpha\rangle \xrightarrow{a} \langle t'_2,\alpha'\rangle$ and $R_{\alpha_0}(\langle t_2,\alpha'\rangle,\langle t'_2,\alpha'\rangle)$.
- *action termination relations*: Suppose $\langle t,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle$. Both $\langle t,\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle$ and $\langle t',\alpha\rangle \xrightarrow{a} \langle \surd,\alpha'\rangle$ do not hold.
- *time step relations*: Suppose $\langle t,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle u,\alpha'\rangle$. There is only one possibility for u, viz. $u \equiv v \cdot t_2$. $\langle t,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle u,\alpha'\rangle$ holds only if $\langle s_1,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle v,\alpha'\rangle$. Because $R^1_{\alpha_0}(\langle s_1,\alpha\rangle,\langle s'_1,\alpha\rangle)$, there exists a v' such that $\langle s'_1,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle v',\alpha'\rangle$ and $R^1_{\alpha_0}(\langle v,\alpha'\rangle,\langle v',\alpha'\rangle)$. So $\langle s'_1 \cdot t'_2,\alpha\rangle \overset{r,\rho}{\longmapsto} \langle v' \cdot t'_2,\alpha'\rangle$ and $R_{\alpha_0}(\langle v \cdot t_2,\alpha'\rangle,\langle v' \cdot t'_2,\alpha'\rangle)$.
- *signal relations*: Suppose $\alpha \in [\mathsf{s}(t)]$. $\alpha \in [\mathsf{s}(t)]$ holds only if $\alpha \in [\mathsf{s}(s_1)]$. Because $R^1_{\alpha_0}(\langle s_1,\alpha\rangle,\langle s'_1,\alpha\rangle)$, we have $\alpha \in [\mathsf{s}(s'_1)]$. So $\alpha \in [\mathsf{s}(s'_1 \cdot t'_2)]$.

Because $R_{\alpha_0}(\langle t_1 \cdot t_2,\alpha_0\rangle,\langle t'_1 \cdot t'_2,\alpha_0\rangle)$, we have that R_{α_0} is a bisimulation relation witnessing $\langle t_1 \cdot t_2,\alpha_0\rangle \leftrightarrows \langle t'_1 \cdot t'_2,\alpha_0\rangle$. Because α_0 is an arbitrary state, we have that there exists a bisimulation relation witnessing $\langle t_1 \cdot t_2,\alpha\rangle \leftrightarrows \langle t'_1 \cdot t'_2,\alpha\rangle$ for any state α. So, we conclude that $t_1 \cdot t_2 \leftrightarrows t'_1 \cdot t'_2$.

A.3 Proof of Theorem 7 (Soundness for $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$*)*

We have to prove that, for all closed terms t and t' of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$, we have $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}} \vdash t = t'$ implies $t \leftrightarrows t'$. It follows from Theorem 6 that it is sufficient to prove for each axiom separately that $t \leftrightarrows t'$ for all closed substitution instances $t = t'$ of the axiom and to prove for each lifting rule, under assumption of the premises of the lifting rule, that $t \leftrightarrows t'$ for all closed substitution instances $t = t'$ of the conclusion of the lifting rule. Moreover, it follows from Lemma 5 that in order to prove that $t \leftrightarrows t'$, it is sufficient to prove that $t \rightleftarrows t'$. It happens that for each axiom of $\mathrm{icBPA}^{\mathrm{srt}}_{\mathrm{hs}}$, we can prove that $t \rightleftarrows t'$ for all closed substitution instances $t = t'$ of the axiom. To prove that $t \rightleftarrows t'$ for all closed substitution instances $t = t'$ of an axiom, we proceed as follows. We give a binary relation R on closed terms and show that (i) for all the closed substitution instances $t = t'$ of the axiom, we have $(t,t') \in R$ and (ii) R is an ic-bisimulation relation.

The proof of (i) is generally trivial. To prove (ii), we show that the conditions for an ic-bisimulation relation are satisfied for all closed terms t^* and t^{**} such that $(t^*, t^{**}) \in R$. We shall loosely say that a relation contains all closed substitution instances of an equation if it contains all pairs (t, t') such that $t = t'$ is a closed substitution instance of the equation.

The axioms of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ are essentially the axioms of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\mathrm{ps}}$ with on top of that axioms NESRU, PSSRU1 and PSSRU2 and the axioms for signal evolution and signal transition given in Table 3. The differences, due to having (undelayable) actions $\tilde{\tilde{a}}$ instead of actions a, are not relevant to the purpose of building on the soundness proofs of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\mathrm{ps}}$.

If we replace in the rules describing the structural operational semantics of $\mathrm{BPA}^{\mathrm{srt}}$ $\_ \xrightarrow{a} \_$, $\_ \xrightarrow{a} \surd$ and $\_ \mapsto \_$ by $\langle \_, \alpha \rangle \xrightarrow{a} \langle \_, \alpha' \rangle$, $\langle \_, \alpha \rangle \xrightarrow{a} \langle \surd, \alpha' \rangle$ and $\langle \_, \alpha \rangle \xmapsto{r,\rho} \langle \_, \alpha' \rangle$, respectively, the induced ic-bisimulation equivalence is identical to the version of bisimulation equivalence for which the axioms of $\mathrm{BPA}^{\mathrm{srt}}$ have been proved sound. If we replace in the rules describing the structural operational semantics of $\mathrm{BPA}_{\mathrm{ps}}$ $\_ \xrightarrow{v,a} \_$, $\_ \xrightarrow{v,a} \surd$, $v \in [\mathsf{s}(\_)]$ and $w \in [\mathsf{s}(\_)]$ by $\langle \_, \alpha \rangle \xrightarrow{a} \langle \_, \alpha' \rangle$, $\langle \_, \alpha \rangle \xrightarrow{a} \langle \surd, \alpha' \rangle$, $\alpha \in [\mathsf{s}(\_)]$ and $\alpha' \in [\mathsf{s}(\_)]$, respectively, the induced ic-bisimulation equivalence is coarser than the version of bisimulation equivalence for which the axioms of $\mathrm{BPA}_{\mathrm{ps}}$ have been proved sound. Hence, we can safely make these replacements.

After that, for some of the operators of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\mathrm{ps}}$, there are still supplementary transition rules concerning additional kinds of transition relations and/or adapted transition rules with supplementary premises concerning additional kinds of transition relations. It follows that, as far as the axioms of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\mathrm{ps}}$ are concerned, we only have to check:

- each axiom in which σ_{rel}, $+$, $\mathbin{:\!\!\rightarrow}$ or $\mathbin{\triangleleft\!\!\cdot}$ occurs with respect to the time step relations;
- each axiom in which σ_{rel} or ν_{rel} occurs with respect to the signal relations.

Checking the axioms concerning σ_{rel} and $+$ with respect to the time step relations goes almost analogous to checking them for $\mathrm{BPA}^{\mathrm{srt}}$: it does not have to be turned upside down in order to take the supplementary premises into account. Checking the axioms concerning $\mathbin{:\!\!\rightarrow}$ and $\mathbin{\triangleleft\!\!\cdot}$ with respect to the time step relations goes analogous to checking them with respect to the action step relations. Checking the axioms concerning σ_{rel} and ν_{rel} with respect to the signal relations is very easy.

Checking axioms NESRU, PSSRU1 and PSSRU2 with respect to all transition relations is very easy as well. What remains, is to check the axioms for signal evolution and signal transition (Table 3) and the lifting rules of $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ (Table 4) with respect to all kinds of transition relations.

For all axioms except axioms HST5 and HST14, and for lifting rule HSELR1, the checks with respect to ic-bisimulation equivalence succeed. In the case of those checks, it happens frequently that for an arbitrary substitution instance $t_1 = t_2$ of an axiom, we can quite easily establish that $\langle t_1, \alpha\rangle \xrightarrow{a} \langle t', \alpha'\rangle$ iff $\langle t_2, \alpha\rangle \xrightarrow{a} \langle t', \alpha'\rangle$, or $\langle t_1, \alpha\rangle \xrightarrow{a} \langle t'_1, \alpha'\rangle$ iff $\langle t_2, \alpha\rangle \xrightarrow{a} \langle t'_2, \alpha'\rangle$ and $t'_1 = t'_2$ is a substitution instance of that axiom as well; and similarly for the other kinds of transition relations.

This is the case except for axioms HSE6, HSE13 and HST6. We present here the checks for axiom HSE13. Checking the other axioms goes similarly, but is simpler.

The checks for axiom HSE13 go as follows. We take the relation R that consists of all closed substitution instances of axiom HSE13, the equation $x = x$ and the equation $\phi \pitchfork_V x + \phi' \pitchfork_{V'} \nu_{\mathsf{rel}}(y) = \phi \pitchfork_V (x + \phi' \pitchfork_{V'} \nu_{\mathsf{rel}}(y))$.

First of all, we consider the closed substitution instances of axiom HSE13. We take an arbitrary closed substitution instance, say $\phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)) = \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)))$, and distinguish between the different kinds of transition relations:

- *action step relations*: For all states α and α', and $a \in \mathsf{A}$, both $\langle \phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)), \alpha\rangle \xrightarrow{a} \langle t', \alpha'\rangle$ and $\langle \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))), \alpha\rangle \xrightarrow{a} \langle t'', \alpha'\rangle$ do not hold for any t' and t''.
- *action termination relations*: For all states α and α', and $a \in \mathsf{A}$, both $\langle \phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$ and $\langle \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$ do not hold.
- *time step relations*: There exist states α and α', $s > 0$ and $\rho \in \mathcal{E}_s$ such that $\langle \phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t', \alpha'\rangle$ or $\langle \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'', \alpha'\rangle$ holds. We proceed by distinguishing three cases:
  - $s = r$: $\langle \phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t', \alpha'\rangle$ holds only if $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\alpha' \in [\mathsf{s}(t_1)]$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_{V'} \phi'$, $\alpha' \in [\mathsf{s}(t_2)]$ and $t' \equiv \phi \pitchfork_V t_1 + \phi' \pitchfork_{V'} \nu_{\mathsf{rel}}(t_2)$. $\langle \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'', \alpha'\rangle$ holds only if $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\alpha' \in [\mathsf{s}(t_1)]$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_{V'} \phi'$, $\alpha' \in [\mathsf{s}(t_2)]$ and $t'' \equiv \phi \pitchfork_V (t_1 + \phi' \pitchfork_{V'} \nu_{\mathsf{rel}}(t_2))$. Moreover, $(t', t'') \in R$.
  - $s < r$: $\langle \phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t', \alpha'\rangle$ holds only if $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_{V'} \phi'$ and $t' \equiv \phi \pitchfork_V \sigma^{r-s}_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^{r-s}_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))$. $\langle \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'', \alpha'\rangle$ holds only if $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_{V'} \phi'$ and $t'' \equiv \phi \pitchfork_V (\sigma^{r-s}_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^{r-s}_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)))$. Moreover, $(t', t'') \in R$.
  - $s > r$: $\langle \phi \pitchfork_V \sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t', \alpha'\rangle$ holds only if $\langle t_1, \alpha''\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t', \alpha'\rangle$ for some state α'', $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\alpha \models \phi'$ and $\alpha \in [\mathsf{s}(t_2)]$. $\langle \phi \pitchfork_V (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \pitchfork_{V'} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'', \alpha'\rangle$ holds only if $\langle t_1, \alpha''\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'', \alpha'\rangle$ for some state α'', $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\alpha \models \phi'$ and

$\alpha \in [\mathsf{s}(t_2)]$. Because there is at most one t^* such that $\langle t_1, \alpha'' \rangle \xmapsto{s-r,\rho} \langle t^*, \alpha' \rangle$, we have that $t' \equiv t''$. Hence, $(t', t'') \in R$.

- *signal relations*: For all states α, $\alpha \in [\mathsf{s}(\phi \mathbin{\curvearrowright_V} \sigma^r_{\mathsf{rel}}(t_1) + \phi' \mathbin{\curvearrowright_{V'}} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2)))]$ holds only if $\alpha \models \phi$ and $\alpha \models \phi'$; and $\alpha \in [\mathsf{s}(\phi \mathbin{\curvearrowright_V} (\sigma^r_{\mathsf{rel}}(t_1) + \phi' \mathbin{\curvearrowright_{V'}} \sigma^r_{\mathsf{rel}}(\nu_{\mathsf{rel}}(t_2))))]$ holds only if $\alpha \models \phi$ and $\alpha \models \phi'$.

Next, we consider the closed substitution instances of the equation $\phi \mathbin{\curvearrowright_V} x + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(y) = \phi \mathbin{\curvearrowright_V} (x + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(y))$. We take an arbitrary closed substitution instance, say $\phi \mathbin{\curvearrowright_V} t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2) = \phi \mathbin{\curvearrowright_V} (t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2))$. It is easy to check that, for all states α and α', $a \in \mathsf{A}$, $r > 0$ and $\rho \in \mathcal{E}_r$, $\langle \phi \mathbin{\curvearrowright_V} t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2), \alpha \rangle \xrightarrow{a} \langle t^*, \alpha' \rangle$ iff $\langle \phi \mathbin{\curvearrowright_V} (t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2)), \alpha \rangle \xrightarrow{a} \langle t^*, \alpha' \rangle$, $\langle \phi \mathbin{\curvearrowright_V} t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2), \alpha \rangle \xrightarrow{a} \langle \surd, \alpha' \rangle$ iff $\langle \phi \mathbin{\curvearrowright_V} (t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2)), \alpha \rangle \xrightarrow{a} \langle \surd, \alpha' \rangle$, $\langle \phi \mathbin{\curvearrowright_V} t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2), \alpha \rangle \xmapsto{r,\rho} \langle t^{**}, \alpha' \rangle$ iff $\langle \phi \mathbin{\curvearrowright_V} (t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2)), \alpha \rangle \xmapsto{r,\rho} \langle t^{**}, \alpha' \rangle$ and $\alpha \in [\mathsf{s}(\phi \mathbin{\curvearrowright_V} t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2))]$ iff $\alpha \in [\mathsf{s}(\phi \mathbin{\curvearrowright_V} (t_1 + \phi' \mathbin{\curvearrowright_{V'}} \nu_{\mathsf{rel}}(t_2)))]$. Moreover, $(t^*, t^*) \in R$ and $(t^{**}, t^{**}) \in R$.

The closed substitution instances of the equation $x = x$ trivially satisfy the conditions for an ic-bisimulation relation.

Axioms HST5 and HST14 and lifting rules HSELR2 and HSELR3 have to be checked with respect to bisimulation equivalence instead of ic-bisimulation equivalence. This goes in a similar way. The differences are that we give a binary relation R on configurations, i.e. pairs of closed terms and states, and show that the conditions for bisimulation equivalence are satisfied. For example, in the case of axiom HST5, we take the relation R that consists of all pairs $(\langle t, \alpha \rangle, \langle t', \alpha \rangle)$ where $t = t'$ is a closed substitution instance of axiom HST5 and α is a state and all pairs $(\langle t^*, \alpha^* \rangle, \langle \chi^\circ \mathbin{\curvearrowright} t^*, \alpha^* \rangle)$ where χ is a transition proposition, t^* is a closed term and α^* is a state such that $\alpha^* \models \chi^\circ$. The restriction on α^* is essential here. It is the reason why checking with respect to ic-bisimulation equivalence fails.

A.4 Proof of Theorem 9 (Elimination for $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$*)*

Like the proof of Theorem 1, the proof is by induction on the structure of closed term t. For terms t of the forms $\bot$, $\tilde{\tilde{a}}$, $\sigma^r_{\mathsf{rel}}(t')$, $t' + t''$, $t' \cdot t''$, $\psi :\to t'$, $\psi \mathbin{\curvearrowright} t'$, $\phi \mathbin{\curvearrowright_V} t'$, $\chi \mathbin{\curvearrowright} t'$ and $\nu_{\mathsf{rel}}(t')$, it follows immediately from the induction hypothesis and Theorem 1, that there is a basic term that is derivably equal to t. For terms of the forms $t' \parallel t''$, $t' \lfloor\!\lfloor t''$, $t' \mid t''$ and $\partial_H(t')$, it follows immediately from the induction hypothesis and the following lemmas:

(1) for all $t, t' \in \mathcal{B}$, there is a $t'' \in \mathcal{B}$ such that $t \parallel t' = t''$ is derivable;
(2) for all $t, t' \in \mathcal{B}$, there is a $t'' \in \mathcal{B}$ such that $t \lfloor\!\lfloor t' = t''$ is derivable;
(3) for all $t, t' \in \mathcal{B}$, there is a $t'' \in \mathcal{B}$ such that $t \mid t' = t''$ is derivable;

(4) for all $t \in \mathcal{B}$, there is a $t' \in \mathcal{B}$ such that $\partial_H(t) = t'$ is derivable.

The fourth lemma is easily proven by induction on the structure of the basic term t. The first three lemmas are proven simultaneously by induction on the sum of the norm of t and the norm of t'. The norm of a closed term t, written $|t|$, is intended to be a measure of the complexity of t. It is defined as follows:

$$
\begin{array}{lll}
|\bot| = |\tilde{\tilde{\delta}}| = |\tilde{\tilde{a}}| = 1\,, & |\psi :\rightarrow t| = |t| + 1\,, & |t \parallel t'| = |t| + |t'| + 1\,, \\
|\sigma^p_{\mathsf{rel}}(t)| = |t| + p + 1\,, & |\psi \mathbin{\curlywedge} t| = |t| + 1\,, & |t \mathbin{\lfloor\!\lfloor} t'| = |t| + |t'| + 1\,, \\
|t + t'| = |t| + |t'| + 1\,, & |\phi \mathbin{\curlyvee_V} t| = |t| + 1\,, & |t \mid t'| = |t| + |t'| + 1\,, \\
|t \cdot t'| = |t| + |t'| + 1\,, & |\chi \mathbin{\curlyvee} t| = |t| + 1\,, & |\partial_H(t)| = |t| + 1\,. \\
|\nu_{\mathsf{rel}}(t)| = |t| + 1\,, &&
\end{array}
$$

The first lemma follows immediately from the second and third lemma. The proof of the second lemma goes by case distinction on the structure of the basic term t, and the proof of the third lemma goes by case distinction on the structure of the basic terms t and t'. We sketch here the proof of the second lemma. The proof of the third lemma is much simpler. The proof of the second lemma is simplified by using a fifth lemma:

(5) for all $t \in \mathcal{B}$, either $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t)) = \bot$ is derivable or there is a $\psi \in {\mathcal{P}_{\mathsf{st}}}^+$ such that $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t)) = \psi \mathbin{\curlywedge} \tilde{\tilde{\delta}}$ is derivable.

This lemma is easily proven by induction on the structure of basic term t.

The proof of the second lemma goes as follows. For the cases $t \equiv \bot$, $t \equiv \psi \mathbin{\curlywedge} \tilde{\tilde{\delta}}$, $t \equiv \psi :\rightarrow (\chi \mathbin{\curlyvee} \tilde{\tilde{a}})$ and $t \equiv \psi :\rightarrow (\chi \mathbin{\curlyvee} \tilde{\tilde{a}} \cdot t^*)$, it is easy to see that a basic term is derivable. The case $t \equiv \psi :\rightarrow (\phi \mathbin{\curlyvee_V} \sigma^r_{\mathsf{rel}}(t^*))$ follows immediately from the fact that

for all $t, t' \in \mathcal{B}$ and $r > 0$, there is a $t'' \in \mathcal{B}$ such that $\sigma^r_{\mathsf{rel}}(t) \mathbin{\lfloor\!\lfloor} t' = t''$ is derivable.

This is proven as follows by case distinction for t' according to Corollary 4:

- $\boldsymbol{t' = \nu_{\mathsf{rel}}(t')}$: Then $\sigma^r_{\mathsf{rel}}(t) \mathbin{\lfloor\!\lfloor} \nu_{\mathsf{rel}}(t') = \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t'))$ by SRCM1aPS. According to the fifth lemma introduced in the proof, either $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t')) = \bot$ is derivable or there is a $\psi \in {\mathcal{P}_{\mathsf{st}}}^+$ such that $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t')) = \psi \mathbin{\curlywedge} \tilde{\tilde{\delta}}$ is derivable; and $\bot, \psi \mathbin{\curlywedge} \tilde{\tilde{\delta}} \in \mathcal{B}$.
- $\boldsymbol{t' = \nu_{\mathsf{rel}}(t') + \sum_{i \in I} \psi_i :\rightarrow (\phi_i \mathbin{\curlyvee_{V_i}} \sigma^{r_i}_{\mathsf{rel}}(t_i))}$: Then $\sigma^r_{\mathsf{rel}}(t) \mathbin{\lfloor\!\lfloor} (\nu_{\mathsf{rel}}(t') + \sum_{i \in I} \psi_i :\rightarrow (\phi_i \mathbin{\curlyvee_{V_i}} \sigma^{r_i}_{\mathsf{rel}}(t_i))) = \sigma^r_{\mathsf{rel}}(t) \mathbin{\lfloor\!\lfloor} \sum_{i \in I} \psi_i :\rightarrow (\phi_i \mathbin{\curlyvee_{V_i}} \sigma^{r_i}_{\mathsf{rel}}(t_i)) + \partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t'))$ by SRCM1bPS. First of all, we look at the term $\sigma^r_{\mathsf{rel}}(t) \mathbin{\lfloor\!\lfloor} \sum_{i \in I} \psi_i :\rightarrow (\phi_i \mathbin{\curlyvee_{V_i}} \sigma^{r_i}_{\mathsf{rel}}(t_i))$, and proceed by distinguishing two cases (we write $r_{\min}$ for $\min(\{r_i \mid i \in I\})$):

- $r > r_{\min}$: Then

$$\begin{aligned}
&\sigma^r_{\mathsf{rel}}(t) \between \sum_{i\in I}\psi_i :\to (\phi_i \lhd_{V_i} \sigma^{r_i}_{\mathsf{rel}}(t_i))\\
&= \sum_{J\subseteq I}(\textstyle\bigwedge_{i\in J}\psi_i \wedge \bigwedge_{i\in I\setminus J}\neg\psi_i) :\to\\
&\quad (\sigma^{r_{\min}}_{\mathsf{rel}}(\sigma^{r-r_{\min}}_{\mathsf{rel}}(t) \between \sum_{i\in J}\phi_i \lhd_{V_i} \sigma^{r_i-r_{\min}}_{\mathsf{rel}}(t_i))\\
&\qquad + (\textstyle\bigwedge_{i\in J}\phi_i) \lhd_{(\bigcup_{i\in J} V_i)} \sigma^{r_{\min}}_{\mathsf{rel}}(\tilde{\tilde{\delta}}))
\end{aligned}$$

  by repeatedly PSSRCM, repeatedly SRT2 and HSSRCM, repeatedly SRT3, and SRCM2. By GC1, HSE1 and the induction hypothesis there is a basic term t^J such that $\sigma^{r-r_{\min}}_{\mathsf{rel}}(t) \between \sum_{i\in J}\phi_i \lhd_{V_i} \sigma^{r_i-r_{\min}}_{\mathsf{rel}}(t_i) = t^J$ for all $J \subseteq I$; and by Theorem 1 there is a basic term t'' such that $\sum_{J\subseteq I}(\bigwedge_{i\in J}\psi_i \wedge \bigwedge_{i\in I\setminus J}\neg\psi_i) :\to (\sigma^{r_{\min}}_{\mathsf{rel}}(t^J) + (\bigwedge_{i\in J}\phi_i) \lhd_{(\bigcup_{i\in J} V_i)} \sigma^{r_{\min}}_{\mathsf{rel}}(\tilde{\tilde{\delta}})) = t''$.
- $r \leq r_{\min}$: Then

$$\begin{aligned}
&\sigma^r_{\mathsf{rel}}(t) \between \sum_{i\in I}\psi_i :\to (\phi_i \lhd_{V_i} \sigma^{r_i}_{\mathsf{rel}}(t_i))\\
&= \sum_{J\subseteq I}(\textstyle\bigwedge_{i\in J}\psi_i \wedge \bigwedge_{i\in I\setminus J}\neg\psi_i) :\to\\
&\quad (\sigma^r_{\mathsf{rel}}(t \between \sum_{i\in J}\phi_i \lhd_{V_i} \sigma^{r_i-r}_{\mathsf{rel}}(t_i)) + (\textstyle\bigwedge_{i\in J}\phi_i) \lhd_{(\bigcup_{i\in J} V_i)} \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}))
\end{aligned}$$

  by repeatedly PSSRCM, repeatedly SRT2 and HSSRCM, repeatedly SRT3, and SRCM2. By GC1, HSE1 and the induction hypothesis there is a basic term t^J such that $t \between \sum_{i\in J}\phi_i \lhd_{V_i} \sigma^{r_i-r}_{\mathsf{rel}}(t_i) = t^J$ for all $J \subseteq I$; and by Theorem 1 there is a basic term t'' such that $\sum_{J\subseteq I}(\bigwedge_{i\in J}\psi_i \wedge \bigwedge_{i\in I\setminus J}\neg\psi_i) :\to (\sigma^r_{\mathsf{rel}}(t^J) + (\bigwedge_{i\in J}\phi_i) \lhd_{(\bigcup_{i\in J} V_i)} \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})) = t''$.

Next, we look at the term $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t'))$. According to the fifth lemma introduced in the proof, either $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t')) = \bot$ is derivable or there is a $\psi \in {\mathcal{P}_{\mathsf{st}}}^+$ such that $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(t')) = \psi \between \tilde{\tilde{\delta}}$ is derivable; and $\bot, \psi \between \tilde{\tilde{\delta}} \in \mathcal{B}$. Hence, t' is in all cases the alternative composition of two basic terms, and thus a basic term.

For the case $t \equiv t^* + t^{**}$, it follows directly from the induction hypothesis and CM4 that a basic term is derivable.

A.5 Proof of Theorem 12 (Soundness for $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$*)*

We have to prove that, for all closed terms t and t' of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$, we have $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}} \vdash t = t'$ implies $t \leftrightarrows t'$. It follows from Theorem 11 that it is sufficient to prove for each axiom separately that $t \leftrightarrows t'$ for all closed substitution instances $t = t'$ of the axiom and to prove for lifting rule HSELR1, under assumption of the

premises of the lifting rule, that $t \leftrightarrows t'$ for all closed substitution instances $t = t'$ of the conclusion of the lifting rule.

The axioms of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$ are essentially the axioms of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$, with the exception of axioms CM2SRPS and CM2SRPS, and on top of that axioms NESRU, PSSRU1 and PSSRU2, axiom PSSRCM, and the axioms for signal evolution and signal transition given in Tables 3, 13 and 14, with the exception of axioms HST5 and HST14. The differences, due to having (undelayable) actions $\tilde{\tilde{a}}$ instead of actions a and using terms $\partial_{\mathsf{A}}(\nu_{\mathsf{rel}}(x))$ instead of $\mathsf{s}_\rho(x) \between \delta$, are not relevant to the purpose of building on the soundness proofs of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$.

In the rules describing the structural operational semantics of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$, we can safely make the replacements mentioned in the soundness proof for $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ (Appendix A.3) as well.

After that, for some of the operators of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$, there are still supplementary transition rules concerning additional kinds of transition relations and/or adapted transition rules with supplementary premises concerning additional kinds of transition relations. It follows that, as far as the axioms of $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$ are concerned, we only have to check:

- each axiom in which $\between$ occurs with respect to the action step and action termination relations;
- each axiom in which σ_{rel}, $+$, $\mathrel{:\!\!\rightarrow}$ or $\between$ occurs with respect to the time step relations;
- each axiom in which σ_{rel}, ν_{rel} or $\between$ occurs with respect to the signal relations;
- each axiom with respect to the discontinuity relations.

For the operators of $\mathrm{icBPA}^{\mathrm{srt}}_{\mathrm{hs}}$, there are, in comparison with the structural operational semantics of $\mathrm{icBPA}^{\mathrm{srt}}_{\mathrm{hs}}$, only supplementary transition rules concerning the discontinuity relations and no adapted transition rules at all. Moreover, the axioms and lifting rule of $\mathrm{icBPA}^{\mathrm{srt}}_{\mathrm{hs}}$ have already been checked with respect to all kinds of transition relations except the discontinuity relations. Hence, as far as the axioms and lifting rule of $\mathrm{icBPA}^{\mathrm{srt}}_{\mathrm{hs}}$ are concerned, we can restrict ourselves to check them with respect to the discontinuity relations. Checking the axioms and lifting rule of $\mathrm{icBPA}^{\mathrm{srt}}_{\mathrm{hs}}$ with respect to the discontinuity relations is easy.

Checking the axioms of $\mathrm{ACP}^{\mathrm{srt}}_{\mathrm{hs}}$ coming from $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$ (other than the axioms of $\mathrm{BPA}^{\mathrm{srt}}$ and $\mathrm{BPA}_{\mathrm{ps}}$), with respect to certain kinds of transition relations as indicated above, goes similar to checking them for $\mathrm{ACP}^{\mathrm{srt}}$ and $\mathrm{ACP}_{\mathrm{ps}}$. Checking axiom PSSRCM is somewhat more complicated, comparable to the checking of axiom HSE13 in the soundness proof for $\mathrm{BPA}^{\mathrm{srt}}_{\mathrm{hs}}$ (Appendix A.3).

What remains, is to check the additional axioms of $\mathrm{ACP}_{\mathrm{hs}}^{\mathrm{srt}}$ concerning signal evolution and signal transition (Table 13) with respect to all kinds of transition relations. Like for most axioms of $\mathrm{BPA}_{\mathrm{hs}}^{\mathrm{srt}}$ concerning signal evolution and signal transition, this is quite easy for most axioms. An exception is axiom HSSRCM.

The checks for axiom HSSRCM go as follows. We take the relation R that consists of all closed substitution instances of axiom HSSRCM, the equation $x = x$, the equation $x \between (\phi \curvearrowright_V y) = x \between (\phi \curvearrowright_V y) + \phi \curvearrowright_V \tilde{\tilde{\delta}}$, the equation $x \between (\phi \curvearrowright_V y + z) = x \between (\phi \curvearrowright_V y + z) + \phi \curvearrowright_V \tilde{\tilde{\delta}}$ and the equation $\sigma^r_{\mathsf{rel}}(x) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(y)) = \sigma^r_{\mathsf{rel}}(x) \between \sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V y) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$.

First, we consider the closed substitution instances of axiom HSSRCM. We take an arbitrary closed substitution instance, say $\sigma^r_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(t_2) + t_3) = \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t_3) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$, and distinguish between the different kinds of transition relations:

- *action step relations*: For all states α and α', and $a \in \mathsf{A}$, both $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(t_2) + t_3), \alpha \rangle \xrightarrow{a} \langle t', \alpha' \rangle$ and $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t_3) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}), \alpha \rangle \xrightarrow{a} \langle t'', \alpha' \rangle$ do not hold for any t' and t''.
- *action termination relations*: For all states α and α', and $a \in \mathsf{A}$, both $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(t_2) + t_3), \alpha \rangle \xrightarrow{a} \langle \surd, \alpha' \rangle$ and $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t_3) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}), \alpha \rangle \xrightarrow{a} \langle \surd, \alpha' \rangle$ do not hold.
- *time step relations*: There exist states α and α', $s > 0$ and $\rho \in \mathcal{E}_s$ such that $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(t_2) + t_3), \alpha \rangle \xmapsto{s,\rho} \langle t', \alpha' \rangle$ or $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t_3) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}), \alpha \rangle \xmapsto{s,\rho} \langle t'', \alpha' \rangle$ holds. We proceed by distinguishing three cases:
  - $s = r$: $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(t_2) + t_3), \alpha \rangle \xmapsto{s,\rho} \langle t', \alpha' \rangle$ holds only if either $\alpha' \in [\mathsf{s}(t_1)]$, $\alpha' \in [\mathsf{s}(t_2)]$, $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \not\xmapsto{s}$, $\alpha \in [\mathsf{s}(t_3)]$ and $t' \equiv t_1 \between (\phi \curvearrowright_V t_2)$ or $\alpha' \in [\mathsf{s}(t_1)]$, $\alpha' \in [\mathsf{s}(t_2)]$, $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \xmapsto{s,\rho} \langle t'_3, \alpha' \rangle$ and $t' \equiv t_1 \between (\phi \curvearrowright_V t_2 + t'_3)$ for some closed term t'_3. $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t_3) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}), \alpha \rangle \xmapsto{s,\rho} \langle t'', \alpha' \rangle$ holds only if either $\alpha' \in [\mathsf{s}(t_1)]$, $\alpha' \in [\mathsf{s}(t_2)]$, $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \not\xmapsto{s}$, $\alpha \in [\mathsf{s}(t_3)]$ and $t'' \equiv t_1 \between (\phi \curvearrowright_V t_2) + \phi \curvearrowright_V \tilde{\tilde{\delta}}$ or $\alpha' \in [\mathsf{s}(t_1)]$, $\alpha' \in [\mathsf{s}(t_2)]$, $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \xmapsto{s,\rho} \langle t'_3, \alpha' \rangle$ and $t'' \equiv t_1 \between (\phi \curvearrowright_V t_2 + t'_3) + \phi \curvearrowright_V \tilde{\tilde{\delta}}$ for some closed term t'_3. If $\langle t_3, \alpha \rangle \not\xmapsto{s}$ and $\alpha \in [\mathsf{s}(t_3)]$, then $(t', t'') \in R$. If $\langle t_3, \alpha \rangle \xmapsto{s,\rho} \langle t'_3, \alpha' \rangle$, then also $(t', t'') \in R$.
  - $s < r$: $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(t_2) + t_3), \alpha \rangle \xmapsto{s,\rho} \langle t', \alpha' \rangle$ holds only if either $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \not\xmapsto{s}$, $\alpha \in [\mathsf{s}(t_3)]$ and $t' \equiv \sigma^{r-s}_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^{r-s}_{\mathsf{rel}}(t_2))$ or $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \xmapsto{s,\rho} \langle t'_3, \alpha' \rangle$ and $t' \equiv \sigma^{r-s}_{\mathsf{rel}}(t_1) \between (\phi \curvearrowright_V \sigma^{r-s}_{\mathsf{rel}}(t_2) + t'_3)$ for some closed term t'_3. $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t_3) + \phi \curvearrowright_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}), \alpha \rangle \xmapsto{s,\rho} \langle t'', \alpha' \rangle$ holds only if either $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \not\xmapsto{s}$, $\alpha \in [\mathsf{s}(t_3)]$ and $t'' \equiv \sigma^{r-s}_{\mathsf{rel}}(t_1) \between \sigma^{r-s}_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + \phi \curvearrowright_V \sigma^{r-s}_{\mathsf{rel}}(\tilde{\tilde{\delta}})$ or $\alpha \xmapsto{s,\rho} \alpha' \models_V \phi$, $\langle t_3, \alpha \rangle \xmapsto{s,\rho} \langle t'_3, \alpha' \rangle$ and $t'' \equiv \sigma^{r-s}_{\mathsf{rel}}(t_1) \between (\sigma^{r-s}_{\mathsf{rel}}(\phi \curvearrowright_V t_2) + t'_3) + \phi \curvearrowright_V \sigma^{r-s}_{\mathsf{rel}}(\tilde{\tilde{\delta}})$ for some closed term t'_3. If $\langle t_3, \alpha \rangle \not\xmapsto{s}$ and $\alpha \in [\mathsf{s}(t_3)]$, then $(t', t'') \in R$. If $\langle t_3, \alpha \rangle \xmapsto{s,\rho} \langle t'_3, \alpha' \rangle$, then

also $(t', t'') \in R$.

  - $s > r$: $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(t_2) + t_3), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t', \alpha'\rangle$ holds only if either $\langle t_1, \alpha^*\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_1, \alpha'\rangle$, $\langle t_2, \alpha^{**}\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_2, \alpha'\rangle$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\langle t_3, \alpha\rangle \stackrel{s}{\not\longmapsto}$, $\alpha \in [\mathsf{s}(t_3)]$ and $t' \equiv t'_1 \between (\phi \between_V t'_2)$ for some closed terms t'_1, t'_2 and states α^* and α^{**}, or $\langle t_1, \alpha^*\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_1, \alpha'\rangle$, $\langle t_2, \alpha^{**}\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_2, \alpha'\rangle$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\langle t_3, \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'_3, \alpha'\rangle$ and $t' \equiv t'_1 \between (\phi \between_V t'_2 + t'_3)$ for some closed terms t'_1, t'_2, t'_3 and states α^* and α^{**}. $\langle \sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \between_V t_2) + t_3) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}), \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'', \alpha'\rangle$ holds only if either $\langle t_1, \alpha^*\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_1, \alpha'\rangle$, $\langle t_2, \alpha^{**}\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_2, \alpha'\rangle$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\langle t_3, \alpha\rangle \stackrel{s}{\not\longmapsto}$, $\alpha \in [\mathsf{s}(t_3)]$ and $t'' \equiv t'_1 \between (\phi \between_V t'_2)$ for some closed terms t'_1, t'_2 and states α^* and α^{**}, or $\langle t_1, \alpha^*\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_1, \alpha'\rangle$, $\langle t_2, \alpha^{**}\rangle \stackrel{s-r,\rho}{\longmapsto} \langle t'_2, \alpha'\rangle$, $\alpha \stackrel{s,\rho}{\longmapsto} \alpha' \models_V \phi$, $\langle t_3, \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'_3, \alpha'\rangle$ and $t'' \equiv t'_1 \between (\phi \between_V t'_2 + t'_3)$ for some closed terms t'_1, t'_2, t'_3 and states α^* and α^{**}. If $\langle t_3, \alpha\rangle \stackrel{s}{\not\longmapsto}$ and $\alpha \in [\mathsf{s}(t_3)]$, then $(t', t'') \in R$. If $\langle t_3, \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t'_3, \alpha'\rangle$, then also $(t', t'') \in R$.
- *signal relations*: For all states α, $\alpha \in [\mathsf{s}(\sigma^r_{\mathsf{rel}}(t_1) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(t_2) + t_3))]$ holds only if $\alpha \models \phi$ and $\alpha \in [\mathsf{s}(t_3)]$; and $\alpha \in [\mathsf{s}(\sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \between_V t_2) + t_3) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}))]$ holds only if $\alpha \models \phi$ and $\alpha \in [\mathsf{s}(t_3)]$.
- *discontinuity relations*: For all states α and α', $\alpha \rightarrow \alpha' \in [\mathsf{d}(\sigma^r_{\mathsf{rel}}(t_1) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(t_2) + t_3))]$ holds only if either $\sigma^r_{\mathsf{rel}}(t_1) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(t_2) + t_3)$ can idle, $\alpha \rightarrow \alpha' \models \mathcal{C}_V$, $\alpha \models \phi$ and $\alpha \rightarrow \alpha' \in [\mathsf{d}(t_3)]$ or $\sigma^r_{\mathsf{rel}}(t_1) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(t_2) + t_3)$ cannot idle and $\alpha \in [\mathsf{s}(t_3)]$; and $\alpha \rightarrow \alpha' \in [\mathsf{d}(\sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \between_V t_2) + t_3) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}}))]$ holds only if $\sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \between_V t_2) + t_3) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$ can idle, $\alpha \rightarrow \alpha' \models \mathcal{C}_V$, $\alpha \models \phi$ and $\alpha \rightarrow \alpha' \in [\mathsf{d}(t_3)]$ or $\sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \between_V t_2) + t_3) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$ cannot idle and $\alpha \in [\mathsf{s}(t_3)]$. Both $\sigma^r_{\mathsf{rel}}(t_1) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(t_2) + t_3)$ and $\sigma^r_{\mathsf{rel}}(t_1) \between (\sigma^r_{\mathsf{rel}}(\phi \between_V t_2) + t_3) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$ can only idle if $\alpha \stackrel{s,\rho}{\longmapsto} \alpha'' \models_V \phi$ or $\langle t_3, \alpha\rangle \stackrel{s,\rho}{\longmapsto} \langle t''_3, \alpha''\rangle$ for some closed term t''_3, state α'', $s > 0$ and $\rho \in \mathcal{E}_s$.

The case of the closed substitution instances of the equation $\sigma^r_{\mathsf{rel}}(x) \between (\phi \between_V \sigma^r_{\mathsf{rel}}(y)) = \sigma^r_{\mathsf{rel}}(x) \between \sigma^r_{\mathsf{rel}}(\phi \between_V y) + \phi \between_V \sigma^r_{\mathsf{rel}}(\tilde{\tilde{\delta}})$ is similar to the previous case.

Next, we consider the closed substitution instances of the equation $x \between (\phi \between_V y + z) = x \between (\phi \between_V y + z) + \phi \between_V \tilde{\tilde{\delta}}$. We take an arbitrary closed substitution instance, say $t_1 \between (\phi \between_V t_2 + t_3) = t_1 \between (\phi \between_V t_2 + t_3) + \phi \between_V \tilde{\tilde{\delta}}$. It is easy to check that, for all states α and α', $a \in \mathsf{A}$, $r > 0$ and $\rho \in \mathcal{E}_r$, $\langle t_1 \between (\phi \between_V t_2 + t_3), \alpha\rangle \xrightarrow{a} \langle t^*, \alpha'\rangle$ iff $\langle t_1 \between (\phi \between_V t_2 + t_3) + \phi \between_V \tilde{\tilde{\delta}}, \alpha\rangle \xrightarrow{a} \langle t^*, \alpha'\rangle$, $\langle t_1 \between (\phi \between_V t_2 + t_3), \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$ iff $\langle t_1 \between (\phi \between_V t_2 + t_3) + \phi \between_V \tilde{\tilde{\delta}}, \alpha\rangle \xrightarrow{a} \langle \surd, \alpha'\rangle$, $\langle t_1 \between (\phi \between_V t_2 + t_3), \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t^{**}, \alpha'\rangle$ iff $\langle t_1 \between (\phi \between_V t_2 + t_3) + \phi \between_V \tilde{\tilde{\delta}}, \alpha\rangle \stackrel{r,\rho}{\longmapsto} \langle t^{**}, \alpha'\rangle$, $\alpha \in [\mathsf{s}(t_1 \between (\phi \between_V t_2 + t_3))]$ iff $\alpha \in [\mathsf{s}(t_1 \between (\phi \between_V t_2 + t_3) + \phi \between_V \tilde{\tilde{\delta}})]$ and $\alpha \rightarrow \alpha' \in [\mathsf{d}(t_1 \between (\phi \between_V t_2 + t_3))]$ iff $\alpha \rightarrow \alpha' \in [\mathsf{d}(t_1 \between (\phi \between_V t_2 + t_3) + \phi \between_V \tilde{\tilde{\delta}})]$. Moreover, $(t^*, t^*) \in R$ and $(t^{**}, t^{**}) \in R$.

The case of the closed substitution instances of the equation $x \between (\phi \between_V y) = x \between (\phi \between_V y) + \phi \between_V \tilde{\tilde{\delta}}$ is similar to the previous case.

The closed substitution instances of the equation $x = x$ trivially satisfy the

conditions for an ic-bisimulation relation.

References

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, P.-H. Ho, Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems, in: R. L. Grossman, A. Nerode, A. P. Ravn, H. Rischel (Eds.), Hybrid Systems, Vol. 736 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 209–229.

[2] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine, The algorithmic analysis of hybrid systems, Theoretical Computer Science 138 (1995) 3–34.

[3] T. A. Henzinger, The theory of hybrid automata, in: LICS'96, IEEE Computer Society Press, 1996, pp. 278–292.

[4] R. Alur, D. L. Dill, Automata for modeling real-time systems, in: M. S. Paterson (Ed.), Proceedings 17th ICALP, Vol. 443 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 322–335.

[5] R. Alur, D. L. Dill, A theory of timed automata, Theoretical Computer Science 126 (1994) 183–235.

[6] R. Alur, T. A. Henzinger, P.-H. Ho, Automatic symbolic verification of embedded systems, IEEE Transactions on Software Engineering 22 (3) (1996) 181–201.

[7] T. A. Henzinger, P.-H. Ho, H. Wong-Toi, HyTech: A model checker for hybrid systems, International Journal on Tools for Technology Transfer 1 (1/2) (1997) 110–122.

[8] T. A. Henzinger, P.-H. Ho, H. Wong-Toi, Algorithmic analysis of nonlinear hybrid systems, IEEE Transactions on Automatic Control 43 (1998) 278–292.

[9] T. A. Henzinger, B. Horowitz, R. Majumdar, H. Wong-Toi, Beyond HyTech: Hybrid systems analysis using interval numerical methods, in: N. Lynch, B. H. Krogh (Eds.), HSCC 2000, Vol. 1790 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 130–144.

[10] J. C. M. Baeten, W. P. Weijland, Process Algebra, Vol. 18 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, 1990.

[11] J. C. M. Baeten, C. A. Middelburg, Process Algebra with Timing, Monographs in Theoretical Computer Science, An EATCS Series, Springer-Verlag, Berlin, 2002.

[12] J. C. M. Baeten, J. A. Bergstra, Process algebra with propositional signals, Theoretical Computer Science 177 (1997) 381–405.

[13] T. A. Henzinger, Z. Manna, A. Pnueli, Towards refining temporal specifications into hybrid systems, in: R. L. Grossman, A. Nerode, A. P. Ravn, H. Rischel (Eds.), Hybrid Systems, Vol. 736 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 60–76.

[14] J. Davies, et al., Timed CSP: Theory and practice, in: J. W. de Bakker, C. Huizing, W. P. de Roever, G. Rozenberg (Eds.), Real Time: Theory and Practice, Vol. 600 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 640–675.

[15] He Jifeng, From CSP to hybrid systems, in: A. W. Roscoe (Ed.), A Classical Mind: Essays in Honour of C. A. R. Hoare, Prentice-Hall, Englewood Cliffs, 1994, pp. 171–189.

[16] R. Milner, Communicating and Mobile Systems: The π-Calculus, Cambridge University Press, Cambridge, 1999.

[17] W. C. Rounds, Hosung Song, The ϕ-calculus: A language for distributed control of reconfigurable embedded systems, in: O. Maler, A. Pnueli (Eds.), HSCC 2003, Vol. 2623 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 435–449.

[18] J. A. Bergstra, C. A. Middelburg, Process algebra for hybrid systems, Computer Science Report 03-06, Department of Mathematics and Computer Science, Eindhoven University of Technology (June 2003).

[19] P. J. L. Cuijpers, M. A. Reniers, Hybrid process algebra, Computer Science Report 03-07, Department of Mathematics and Computer Science, Eindhoven University of Technology (July 2003).

[20] R. Alur, R. Grosu, I. Lee, O. Sokolsky, Compositional refinement for hierarchical hybrid systems, in: M. D. Di Benedetto, A. Sangiovanni-Vincentelli (Eds.), HSCC 2001, Vol. 2034 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 33–48.

[21] T. A. Henzinger, Assume-guarantee reasoning for hierarchical hybrid systems, in: M. D. Di Benedetto, A. Sangiovanni-Vincentelli (Eds.), HSCC 2001, Vol. 2034 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 275–290.

[22] N. Lynch, R. Segala, F. W. Vaandrager, Hybrid I/O automata, Information and Computation 185 (1) (2003) 105–157.

[23] J. A. Bergstra, J. W. Klop, The algebra of recursively defined processes and the algebra of regular processes, in: J. Paredaens (Ed.), Proceedings 11th ICALP, Vol. 172 of Lecture Notes in Computer Science, Springer-Verlag, 1984, pp. 82–95.

[24] J. C. M. Baeten, C. A. Middelburg, Process algebra with timing: Real time and discrete time, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, pp. 627–684.

[25] C. A. Middelburg, Process algebra with nonstandard timing, Fundamenta Informaticae 53 (1) (2002) 55–77.

[26] C. A. Middelburg, Revisiting timing in process algebra, Journal of Logic and Algebraic Programming 54 (1/2) (2003) 109–127.

[27] A. Browder, Mathematical Analysis: An Introduction, Springer-Verlag, Berlin, 1996.

[28] X. Nicollin, J. Sifakis, The algebra of timed processes ATP: Theory and application, Information and Computation 114 (1) (1994) 131–178.

[29] L. Chen, An interleaving model for real-time systems, in: A. Nerode, M. Taitslin (Eds.), Symposium on Logical Foundations of Computer Science, Vol. 620 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 81–92.

[30] F. Moller, C. Tofts, A temporal calculus of communicating systems, in: J. C. M. Baeten, J. W. Klop (Eds.), CONCUR'90, Vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 401–415.

[31] Wang Yi, Real-time behaviour of asynchronous agents, in: J. C. M. Baeten, J. W. Klop (Eds.), CONCUR'90, Vol. 458 of Lecture Notes in Computer Science, Springer-Verlag, 1990, pp. 502–520.

[32] J. Quemada, D. de Frutos, A. Azcorra, TIC: A timed calculus, Formal Aspects of Computing 5 (3) (1993) 224–252.

[33] M. Hennessy, T. Regan, A process algebra for timed systems, Information and Computation 117 (1995) 221–239.

[34] J. A. Bergstra, C. A. Middelburg, Y. S. Usenko, Discrete time process algebra and the semantics of SDL, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, pp. 1209–1268.

[35] C. A. Middelburg, Variable binding operators in transition system specifications, Journal of Logic and Algebraic Programming 47 (1) (2001) 15–45.

[36] C. A. Middelburg, An alternative formulation of operational conservativity with binding terms, Journal of Logic and Algebraic Programming 55 (1/2) (2003) 1–19.

[37] J. F. Groote, J. J. van Wamel, Analysis of three hybrid systems in timed μCRL, Science of Computer Programming 39 (2/3) (2001) 215–247.

[38] J. A. Bergstra, J. W. Klop, Verification of an alternating bit protocol by means of process algebra, in: W. Bibel, K. P. Jantke (Eds.), Mathematical Methods of Specification and Synthesis of Software Systems, Vol. 215 of Lecture Notes in Computer Science, Springer-Verlag, 1986, pp. 9–23.

[39] J. F. Groote, A. Ponse, Process algebra with guards: Combining Hoare logic with process algebra, Formal Aspects of Computing 6 (2) (1994) 115–164.

[40] L. Aceto, W. J. Fokkink, C. Verhoef, Structural operational semantics, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, pp. 197–292.

[41] J. A. Bergstra, W. J. Fokkink, C. A. Middelburg, Algebra of timed frames, International Journal of Computer Mathematics 61 (1996) 227–255.

[42] C. A. Middelburg, Truth of duration calculus formulae in timed frames, Fundamenta Informaticae 36 (2/3) (1998) 235–263.

[43] J. J. Vereijken, A process algebra for hybrid systems, extended abstract of talk presented at Second European Workshop on Real-Time and Hybrid Systems, Grenoble, France, 1995 (1995).

[44] J. A. Bergstra, C. A. Middelburg, Continuity controlled hybrid automata, Computer Science Report 04-11, Department of Mathematics and Computer Science, Eindhoven University of Technology (April 2004).

[45] R. J. van Glabbeek, The linear time – branching time spectrum I, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, Elsevier, Amsterdam, 2001, pp. 3–99.

[46] Zhou Chaochen, C. A. R. Hoare, A. P. Ravn, A calculus of durations, Information Processing Letters 40 (1991) 269–276.

[47] Zhou Chaochen, A. P. Ravn, M. R. Hansen, An extended duration calculus for hybrid real-time systems, in: R. L. Grossman, A. Nerode, A. P. Ravn, H. Rischel (Eds.), Hybrid Systems, Vol. 736 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 36–59.

[48] J. C. M. Baeten, J. A. Bergstra, M. A. Reniers, Discrete time process algebra with silent step, in: G. D. Plotkin, C. Stirling, M. Tofte (Eds.), Proof, Language and Interaction: Essays in Honour of Robin Milner, MIT Press, Cambridge, MA, 2000, pp. 535–569.

[49] J. C. M. Baeten, C. A. Middelburg, M. A. Reniers, A new equivalence for processes with timing – with an application to protocol verification, Computer Science Report 02-10, Department of Mathematics and Computer Science, Eindhoven University of Technology (October 2002).

[50] P. J. L. Cuijpers, M. A. Reniers, Topological (bi-)simulation, Computer Science Report 02-04, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven (April 2001).

[51] J. C. M. Baeten, J. A. Bergstra, Real time process algebra with infinitesimals, in: A. Ponse, C. Verhoef, S. F. M. van Vlijmen (Eds.), Algebra of Communicating Processes 1994, Workshops in Computing Series, Springer-Verlag, 1995, pp. 148–187.

[52] J. C. M. Baeten, J. A. Bergstra, Real time process algebra, Formal Aspects of Computing 3 (2) (1991) 142–188.

[53] J. C. M. Baeten, J. A. Bergstra, Real space process algebra, Formal Aspects of Computing 5 (6) (1993) 481–529.

[54] J. A. Bergstra, C. A. Middelburg, Located actions in process algebra with timing, to appear in *Fundamenta Informaticae*, 2004.

[55] J. J. Vereijken, Discrete time process algebra, Ph.D. thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven (1997).