Revisiting Timing in Process Algebra

C.A. Middelburg*

Computing Science Department, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, the Netherlands

Department of Philosophy, Utrecht University
P.O. Box 80126, 3508 TC Utrecht, the Netherlands

mailto:keesm@win.tue.nl, http://www.win.tue.nl/~keesm

Abstract

We shortly review the framework of process algebras with timing presented by Baeten and Middelburg [Handbook of Process Algebra, Elsevier, 2001, Chapter 10]. In order to cover processes that are capable of performing certain actions at all points in some time interval, we add integration to the process algebra with continuous relative timing from this framework. This extension happens to reveal some points that are peculiar to relative timing. We go into these points. The most flagrant point is that, unlike in case of absolute timing, discretization cannot be added to the extension without first adding a mechanism for parametric timing like initial abstraction.

Keywords: process algebra, relative timing, absolute timing, parametric timing, discrete time scale, continuous time scale, integration, initial abstraction, discretization.

1994 CR Categories: D.1.3, D.3.1, F.1.2, F.3.1.

1 Introduction

In [6], a coherent collection of four algebraic theories about processes, each dealing with timing in a different way, is introduced. The timing of actions is either relative (to the time at which the preceding action is performed) or absolute and the time scale on which the time is measured is either discrete or continuous. The theories concerned are extensions of ACP [8, 9, 10] which originate mainly from the work on process algebra with timing presented in [1, 2, 3, 4, 5].

Process algebras with relative timing are generally considered simpler than ones with absolute timing. Nearly all process algebras with timing offer relative timing, e.g. the different versions of CCS with timing [12, 15, 18], Timed CSP [13], TPL [14], ATP [16] and TIC [17]. Experience of applying the versions of ACP with timing presented in [6], e.g. to data communication protocols, a

*The work presented in this paper has been partly carried out while the author was at Eindhoven Embedded Systems Institute, Eindhoven University of Technology.

mutual exclusion protocol and controllers of various systems, was acquired during the preparation of [7]. It has shown that the versions with absolute timing are not more difficult to use in describing and analysing the time-dependent behaviour of systems. Additionally, further work on the framework introduced in [6] has revealed some points that are peculiar to relative timing and as a matter of fact shortcomings of relative timing.

In this paper, we will first shortly review the framework of process algebras with timing introduced in [6]. This covers the four principal process algebras (Section 2), the addition of the integration operator to the process algebra with continuous absolute timing (Section 3) and the addition of a mechanism for parametric timing to both process algebras with absolute timing (Section 4). Next, we will go into the following points peculiar to relative timing:

- the rules for the operational semantics concerning the integration operator are far more intricate than the ones in case of absolute timing (Section 5);
- the operational semantics of the extension with a time-dependent state operator requires an operational semantics as detailed as in case of absolute timing (Section 6);
- the version of ACP with discrete relative timing cannot be embedded in the version with continuous relative timing (Section 7);
- the discretization operator cannot be added to the version of ACP with continuous relative timing (Section 8).

Acknowledgements The author thanks Jos Baeten for pointing out to me that the version of ACP with discrete relative timing cannot be embedded in the version with continuous relative timing.

2 Principal process algebras with timing

In this section, we give an overview of the four principal versions of process algebra with timing introduced in [6]. In these versions, execution of actions and passage of time are separated. Versions in which execution of actions and passage of time are combined can simply be regarded as specializations.

Measuring time on a discrete time scale does not mean that the execution of actions is restricted to discrete points in time. It means that time is divided into time slices and timing of actions is done with respect to the time slices in which they are performed: if an action can be performed in time slice $n + 1$, it can be performed at any time $r \in \mathbb{R}^{+}$ such that $n \leq r < n + 1$. Thus, the versions of ACP with timing where time is measured on a discrete time scale permit to consider systems at a more abstract level than the versions where time is measured on a continuous time scale, a level where time is measured with finite precision.

The possibility of two or more actions to be performed consecutively at the same point of time is not excluded. This urgency is useful in practice when describing and analysing systems in which actions occur that are entirely independent, such as actions that happen at different locations in a distributed system.

Timing with respect to points of time on a continuous time scale is generally considered to be the standard way of timing. Therefore, the versions with continuous relative timing and continuous absolute timing are alternatively called the versions with standard relative timing and standard absolute timing.

We only consider the basic process algebras, which are the subtheories that do not cover parallelism and communication, because they are sufficient for the points we want to make. In the case of ACP, the basic process algebra is called BPA_δ. BPA is the subtheory of BPA_δ that does not cover deadlock. In [6], models for the axioms of the theories considered are presented using structural operational semantics and bisimulation. For each of the theories concerned, the axioms form a sound and (relative) complete axiomatization of the corresponding model. Besides, under a mild syntactic restriction on the defining equations of recursively defined processes, known as guardedness, the defining equations have unique solutions in these models.

2.1 Discrete relative timing

The atomic processes are undelayable actions. Let a be an action. Then *undelayable action* a, written $\underline{\underline{a}}$, is the process that performs action a in the current time slice and then terminates successfully. Undelayable 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 $n \in \mathbb{N}$. Then the *relative delay* of p for n time slices, written $\sigma^n_{\mathsf{rel}}(p)$, is the process that idles till the n^{th} next time slice and then behaves like p. In other words, it is p after a delay of n time slices.

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 till the next time slice. This process, written $\underline{\underline{\delta}}$, is called *undelayable deadlock*. In order to handle situations in which processes exhibit inconsistent timing, it is preferable to have an additional process that can be viewed as (a trace of) a process that has deadlocked before the current time slice. This process, written $\dot{\delta}$, is called the *deadlocked process*. The deadlocked process after a delay of one time slice and undelayable deadlock are considered to be indistinguishable from each other.

In order to capture timing fully, we have, in addition to relative delay, relative time-out and relative initialization. Let p be a process and $n \in \mathbb{N}$. The *relative time-out* of p after n time slices, written $\upsilon^n_{\mathsf{rel}}(p)$, behaves either like the part of p that does not idle till the n^{th} next time slice or like the deadlocked process after a delay of n time slices if p is capable of idling till that time slice. Otherwise, it behaves like p. The *relative initialization* of p after n time slices, written $\overline{\upsilon}^n_{\mathsf{rel}}(p)$, behaves like the part of p that idles till the n^{th} next time slice if p is capable of idling till that time slice. Otherwise, it behaves like the deadlocked process after a delay of n time slices.

We use the sum notation $\sum_{i \in \mathcal{I}} t_i$, where $\mathcal{I} = \{i_1, \ldots, i_n\}$, for the alternative composition $t_{i_1} + \ldots + t_{i_n}$.

Axioms of BPA$^{\mathrm{drt}}$ The axiom system of BPA$^{\mathrm{drt}}$ consists of the equations given in Table 1.

$x + y = y + x$	A1	$\sigma^0_{\mathsf{rel}}(x) = x$	DRT1
$(x + y) + z = x + (y + z)$	A2	$\sigma^m_{\mathsf{rel}}(\sigma^n_{\mathsf{rel}}(x)) = \sigma^{m+n}_{\mathsf{rel}}(x)$	DRT2
$x + x = x$	A3	$\sigma^n_{\mathsf{rel}}(x) + \sigma^n_{\mathsf{rel}}(y) = \sigma^n_{\mathsf{rel}}(x + y)$	DRT3
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4	$\sigma^n_{\mathsf{rel}}(x) \cdot y = \sigma^n_{\mathsf{rel}}(x \cdot y)$	DRT4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5	$\sigma^1_{\mathsf{rel}}(\dot{\delta}) = \underline{\underline{\delta}}$	DRT7
$x + \dot{\delta} = x$	A6ID	$\underline{\underline{a}} + \underline{\underline{\delta}} = \underline{\underline{a}}$	A6DRa
$\dot{\delta} \cdot x = \dot{\delta}$	A7ID		
$\upsilon^n_{\mathsf{rel}}(\dot{\delta}) = \dot{\delta}$	DRTO0	$\overline{\upsilon}^n_{\mathsf{rel}}(\dot{\delta}) = \sigma^n_{\mathsf{rel}}(\dot{\delta})$	DRI0
$\upsilon^0_{\mathsf{rel}}(x) = \dot{\delta}$	DRTO1	$\overline{\upsilon}^0_{\mathsf{rel}}(x) = x$	DRI1
$\upsilon^{n+1}_{\mathsf{rel}}(\underline{\underline{a}}) = \underline{\underline{a}}$	DRTO2	$\overline{\upsilon}^{n+1}_{\mathsf{rel}}(\underline{\underline{a}}) = \sigma^n_{\mathsf{rel}}(\underline{\underline{\delta}})$	DRI2
$\upsilon^{m+n}_{\mathsf{rel}}(\sigma^n_{\mathsf{rel}}(x)) = \sigma^n_{\mathsf{rel}}(\upsilon^m_{\mathsf{rel}}(x))$	DRTO3	$\overline{\upsilon}^{m+n}_{\mathsf{rel}}(\sigma^n_{\mathsf{rel}}(x)) = \sigma^n_{\mathsf{rel}}(\overline{\upsilon}^m_{\mathsf{rel}}(x))$	DRI3
$\upsilon^n_{\mathsf{rel}}(x + y) = \upsilon^n_{\mathsf{rel}}(x) + \upsilon^n_{\mathsf{rel}}(y)$	DRTO4	$\overline{\upsilon}^n_{\mathsf{rel}}(x + y) = \overline{\upsilon}^n_{\mathsf{rel}}(x) + \overline{\upsilon}^n_{\mathsf{rel}}(y)$	DRI4
$\upsilon^n_{\mathsf{rel}}(x \cdot y) = \upsilon^n_{\mathsf{rel}}(x) \cdot y$	DRTO5	$\overline{\upsilon}^n_{\mathsf{rel}}(x \cdot y) = \overline{\upsilon}^n_{\mathsf{rel}}(x) \cdot y$	DRI5

Table 1: Axioms of BPA$^{\mathrm{drt}}$ ($a \in \mathsf{A}_\delta$, $m, n \geq 0$)

Axioms A1-A5 are the axioms of BPA. Axioms A6ID and A7ID are simple reformulations of axioms A6 and A7 of BPA$_\delta$. Axioms DRT1 and DRT2 point out that a delay of 0 time slices has no effect and that consecutive delays count up. Axiom DRT3, called the time-factorization axiom, shows that a delay by itself cannot determine a choice. Axiom DRT4 reflects that timing is relative. This axiom makes the equation $\sigma^n_{\mathsf{rel}}(\dot{\delta}) \cdot x = \sigma^n_{\mathsf{rel}}(\dot{\delta})$ (DRT6) derivable. Axiom DRT7 expresses that the deadlocked process after a delay of one time slice cannot be distinguished from undelayable deadlock. This axiom makes the equations $\sigma^{n+1}_{\mathsf{rel}}(x) + \underline{\underline{\delta}} = \sigma^{n+1}_{\mathsf{rel}}(x)$ (A6DRb) and $\underline{\underline{\delta}} \cdot x = \underline{\underline{\delta}}$ (A7DR) derivable. The equation $t + \underline{\underline{\delta}} = t$ is only derivable for closed terms $t \neq \dot{\delta}$. Axioms DRTO0-DRTO5 and DRI0-DRI5 are the defining equations of the relative time-out operator and relative initialization operator, respectively. These axioms reflect the intended meaning of these operators clearly.

Example 1 We consider a process that polls on two input ports (ports 1 and 2) by repeatedly enabling each of them in turn for 1 millisecond. When a datum d is offered at a port while it is enabled, the polling process delivers it at its single output port (port 3). It is assumed that the set D of all possible data is finite. This polling process can be recursively defined in BPA$^{\mathrm{drt}}$ as follows:

$$Poll = \sum_{d \in D} \underline{\underline{r_1(d)}} \cdot \underline{\underline{s_3(d)}} + \sigma^1_{\mathsf{rel}}(\sum_{d \in D} \underline{\underline{r_2(d)}} \cdot \underline{\underline{s_3(d)}}) + \sigma^2_{\mathsf{rel}}(Poll)$$

Notice that we are not able to describe that an accepted datum is immediately delivered.

2.2 Discrete absolute timing

The constants and operators of BPA$^{\mathrm{dat}}$ differ from the ones of BPA$^{\mathrm{drt}}$ as follows. In BPA$^{\mathrm{dat}}$, we have the constants $\underline{a}$ and $\underline{\delta}$ instead of $\underline{\underline{a}}$ and $\underline{\underline{\delta}}$, and the operator σ_{abs} (absolute delay) instead of σ_{rel} (relative delay). The constant $\underline{a}$ stands for the process that performs a in the first time slice and then terminates successfully. The constant $\underline{\delta}$ stands for the process that is neither capable of performing any action nor capable of idling till after the first time slice. We also have absolute counterparts of the relative time-out and initialization operators: υ_{abs} (absolute time-out) and $\overline{\upsilon}_{\mathsf{abs}}$ (absolute initialization). The deadlocked process can now be viewed as a process that has deadlocked before the first time slice. There is no reason to distinguish it from the deadlocked process in the case of relative timing, as explained in Section 7.

The operator υ_{abs} makes it relatively easy to state the fundamental facts concerning the interaction of absolute delay with sequential composition in the form of equational axioms. The operator $\overline{\upsilon}_{\mathsf{abs}}$ is used to anticipate in the formulation of these facts the addition of a mechanism for parametric timing by which a process cannot only to be started up at time 0, but also at other discrete points of time. Without this parametrization mechanism, we have for any process p that the absolute initialization of p at time 0 behaves like p.

Axioms of BPA$^{\mathrm{dat}}$ The axiom system of BPA$^{\mathrm{dat}}$ consists of axioms A1-A5, A6ID and A7ID from Table 1 and the equations given in Table 2.

Axioms DAT1-DAT3, DAT7, A6DAa and DATO0-DATO5 are simple reformulations of axioms DRT1-DRT3, DRT7, A6DRa and DRTO0-DRTO5 of BPA$^{\mathrm{drt}}$. Instead of axioms DAI0-DAI5, we could have taken simple reformulations of axioms DRI0-DRI5. However, those alternative axioms do not accomodate the addition of the mechanism for parametric timing mentioned before. Striking is the replacement of axiom DRT4 by the axioms DAT4 and DAT5 as well as the addition of axiom DAT6. Axioms DAT4 and DAT5 reflect that timing is absolute. These axioms become easier to understand by realizing that for all closed BPA$^{\mathrm{dat}}$-terms t and for all $n > 0$ either there exists a closed term t' such that $t = \upsilon^n_{\mathsf{abs}}(t')$ is derivable or there exist closed terms t' and t'' such that $t = \upsilon^n_{\mathsf{abs}}(t') + \sigma^n_{\mathsf{abs}}(t'')$ is derivable. Besides, $\overline{\upsilon}^0_{\mathsf{abs}}(t) = t$ is derivable for all closed BPA$^{\mathrm{dat}}$-terms t. Unlike its counterpart in BPA$^{\mathrm{drt}}$, axiom DAT6 is not derivable. The cause of this is the absence of a true counterpart of axiom DRT4.

$\sigma^0_{\mathsf{abs}}(x) = \overline{\upsilon}^0_{\mathsf{abs}}(x)$	DAT1	$\sigma^n_{\mathsf{abs}}(\dot{\delta}) \cdot x = \sigma^n_{\mathsf{abs}}(\dot{\delta})$	DAT6
$\sigma^m_{\mathsf{abs}}(\sigma^n_{\mathsf{abs}}(x)) = \sigma^{m+n}_{\mathsf{abs}}(x)$	DAT2	$\sigma^1_{\mathsf{abs}}(\dot{\delta}) = \underline{\delta}$	DAT7
$\sigma^n_{\mathsf{abs}}(x) + \sigma^n_{\mathsf{abs}}(y) = \sigma^n_{\mathsf{abs}}(x+y)$	DAT3	$\underline{a} + \underline{\delta} = \underline{a}$	A6DAa
$\sigma^n_{\mathsf{abs}}(x) \cdot \upsilon^n_{\mathsf{abs}}(y) = \sigma^n_{\mathsf{abs}}(x \cdot \dot{\delta})$	DAT4		
$\sigma^n_{\mathsf{abs}}(x) \cdot (\upsilon^n_{\mathsf{abs}}(y) + \sigma^n_{\mathsf{abs}}(z)) = \sigma^n_{\mathsf{abs}}(x \cdot \overline{\upsilon}^0_{\mathsf{abs}}(z))$	DAT5		
		$\overline{\upsilon}^0_{\mathsf{abs}}(\dot{\delta}) = \dot{\delta}$	DAI0a
$\upsilon^n_{\mathsf{abs}}(\dot{\delta}) = \dot{\delta}$	DATO0	$\overline{\upsilon}^{n+1}_{\mathsf{abs}}(\dot{\delta}) = \sigma^{n+1}_{\mathsf{abs}}(\dot{\delta})$	DAI0b
$\upsilon^0_{\mathsf{abs}}(x) = \dot{\delta}$	DATO1	$\overline{\upsilon}^0_{\mathsf{abs}}(\underline{a}) = \underline{a}$	DAI1
$\upsilon^{n+1}_{\mathsf{abs}}(\underline{a}) = \underline{a}$	DATO2	$\overline{\upsilon}^{n+1}_{\mathsf{abs}}(\underline{a}) = \sigma^{n+1}_{\mathsf{abs}}(\dot{\delta})$	DAI2
$\upsilon^{m+n}_{\mathsf{abs}}(\sigma^n_{\mathsf{abs}}(x)) = \sigma^n_{\mathsf{abs}}(\upsilon^m_{\mathsf{abs}}(x))$	DATO3	$\overline{\upsilon}^{m+n}_{\mathsf{abs}}(\sigma^n_{\mathsf{abs}}(x)) = \sigma^n_{\mathsf{abs}}(\overline{\upsilon}^m_{\mathsf{abs}}(\overline{\upsilon}^0_{\mathsf{abs}}(x)))$	DAI3
$\upsilon^n_{\mathsf{abs}}(x+y) = \upsilon^n_{\mathsf{abs}}(x) + \upsilon^n_{\mathsf{abs}}(y)$	DATO4	$\overline{\upsilon}^n_{\mathsf{abs}}(x+y) = \overline{\upsilon}^n_{\mathsf{abs}}(x) + \overline{\upsilon}^n_{\mathsf{abs}}(y)$	DAI4
$\upsilon^n_{\mathsf{abs}}(x \cdot y) = \upsilon^n_{\mathsf{abs}}(x) \cdot y$	DATO5	$\overline{\upsilon}^n_{\mathsf{abs}}(x \cdot y) = \overline{\upsilon}^n_{\mathsf{abs}}(x) \cdot y$	DAI5

Table 2: Axioms of $\mathrm{BPA}^{\mathrm{dat}}$ ($a \in \mathsf{A}_\delta$, $m, n \geq 0$)

Example 2 We consider the polling process of Example 1 again. It can be recursively defined in $\mathrm{BPA}^{\mathrm{dat}}$ as follows:

$$Poll = \sum_{d \in D} \underline{r_1(d)} \cdot \underline{s_3(d)} + \sigma^1_{\mathsf{abs}}(\sum_{d \in D} \underline{r_2(d)} \cdot \underline{s_3(d)}) + \sigma^2_{\mathsf{abs}}(Poll)$$

If the process *Poll* from Example 1 is started up at time 0, that process behaves exactly the same as the process defined above.

2.3 Continuous relative timing

The constants and operators of $\mathrm{BPA}^{\mathrm{srt}}$ differ from the ones of $\mathrm{BPA}^{\mathrm{drt}}$ as follows. In $\mathrm{BPA}^{\mathrm{srt}}$, we have the constants $\tilde{\tilde{a}}$ and $\tilde{\tilde{\delta}}$ instead of $\underline{\underline{a}}$ and $\underline{\underline{\delta}}$. The constant $\tilde{\tilde{a}}$ stands for the proces that performs a at the current point of time and then terminates succesfully. The constant $\tilde{\tilde{\delta}}$ stand for the process that is neither capable of performing any action nor capable of idling till after the current point of time. The operators σ_{rel}, υ_{rel} and $\overline{\upsilon}_{\mathsf{rel}}$ have a non-negative real number instead of a natural number as their first argument. The deadlocked process can now be viewed as a process that has deadlocked before the current point of time. We do not bother about distinguishing it from the deadlocked process in the case that time is measured on a discrete time scale for reasons that become clear later in Section 7.

Axioms of $\mathrm{BPA}^{\mathrm{srt}}$ The axiom system of $\mathrm{BPA}^{\mathrm{srt}}$ consists of axioms A1-A5, A6ID and A7ID from Table 1 and the equations given in Table 3.

Axioms SRT1-SRT4, A6SRa, SRTO0-SRTO5 and SRI0-SRI5 are simple reformulations of axioms DRT1-DRT4, A6DRa, DRTO0-DRTO5 and DRI0-DRI5 of $\mathrm{BPA}^{\mathrm{drt}}$. Unlike their counterparts in $\mathrm{BPA}^{\mathrm{drt}}$, axioms A6SRb and A7SR are not derivable. The cause of this is the absence of a counterpart of axiom DRT7.

$\sigma^{0}_{\mathsf{rel}}(x) = x$	SRT1	$\tilde{\tilde{a}} + \tilde{\tilde{\delta}} = \tilde{\tilde{a}}$	A6SRa
$\sigma^{p}_{\mathsf{rel}}(\sigma^{q}_{\mathsf{rel}}(x)) = \sigma^{p+q}_{\mathsf{rel}}(x)$	SRT2	$\sigma^{r}_{\mathsf{rel}}(x) + \tilde{\tilde{\delta}} = \sigma^{r}_{\mathsf{rel}}(x)$	A6SRb
$\sigma^{p}_{\mathsf{rel}}(x) + \sigma^{p}_{\mathsf{rel}}(y) = \sigma^{p}_{\mathsf{rel}}(x+y)$	SRT3	$\tilde{\tilde{\delta}} \cdot x = \tilde{\tilde{\delta}}$	A7SR
$\sigma^{p}_{\mathsf{rel}}(x) \cdot y = \sigma^{p}_{\mathsf{rel}}(x \cdot y)$	SRT4		
$\upsilon^{p}_{\mathsf{rel}}(\dot{\delta}) = \dot{\delta}$	SRTO0	$\overline{\upsilon}^{p}_{\mathsf{rel}}(\dot{\delta}) = \sigma^{p}_{\mathsf{rel}}(\dot{\delta})$	SRI0
$\upsilon^{0}_{\mathsf{rel}}(x) = \dot{\delta}$	SRTO1	$\overline{\upsilon}^{0}_{\mathsf{rel}}(x) = x$	SRI1
$\upsilon^{r}_{\mathsf{rel}}(\tilde{\tilde{a}}) = \tilde{\tilde{a}}$	SRTO2	$\overline{\upsilon}^{r}_{\mathsf{rel}}(\tilde{\tilde{a}}) = \sigma^{r}_{\mathsf{rel}}(\dot{\delta})$	SRI2
$\upsilon^{p+q}_{\mathsf{rel}}(\sigma^{p}_{\mathsf{rel}}(x)) = \sigma^{p}_{\mathsf{rel}}(\upsilon^{q}_{\mathsf{rel}}(x))$	SRTO3	$\overline{\upsilon}^{p+q}_{\mathsf{rel}}(\sigma^{p}_{\mathsf{rel}}(x)) = \sigma^{p}_{\mathsf{rel}}(\overline{\upsilon}^{q}_{\mathsf{rel}}(x))$	SRI3
$\upsilon^{p}_{\mathsf{rel}}(x+y) = \upsilon^{p}_{\mathsf{rel}}(x) + \upsilon^{p}_{\mathsf{rel}}(y)$	SRTO4	$\overline{\upsilon}^{p}_{\mathsf{rel}}(x+y) = \overline{\upsilon}^{p}_{\mathsf{rel}}(x) + \overline{\upsilon}^{p}_{\mathsf{rel}}(y)$	SRI4
$\upsilon^{p}_{\mathsf{rel}}(x \cdot y) = \upsilon^{p}_{\mathsf{rel}}(x) \cdot y$	SRTO5	$\overline{\upsilon}^{p}_{\mathsf{rel}}(x \cdot y) = \overline{\upsilon}^{p}_{\mathsf{rel}}(x) \cdot y$	SRI5

Table 3: Axioms of $\mathrm{BPA}^{\mathrm{srt}}$ ($a \in \mathsf{A}_{\delta}$, $p, q \geq 0$, $r > 0$)

2.4 Continuous absolute timing

The constants and operators of $\mathrm{BPA}^{\mathrm{sat}}$ differ from the ones of $\mathrm{BPA}^{\mathrm{dat}}$ as follows. In $\mathrm{BPA}^{\mathrm{sat}}$, we have the constants $\tilde{a}$ and $\tilde{\delta}$ instead of $\underline{a}$ and $\underline{\delta}$. The constant $\tilde{a}$ stands for the proces that performs a at point of time 0 and then terminates succesfully. The constant $\tilde{\delta}$ stand for the process that is neither capable of performing any action nor capable of idling till after point of time 0. The operators σ_{abs}, υ_{abs} and $\overline{\upsilon}_{\mathsf{abs}}$ have a non-negative real number instead of a natural number as their first argument. The deadlocked process can now be viewed as a process that has deadlocked before point of time 0. There is no reason to distinguish it from the deadlocked process in the case that time is measured on a discrete time scale. This can be explained as follows: a process has deadlocked before the first time slice if and only if it has deadlocked before point of time 0.

Axioms of $\mathrm{BPA}^{\mathrm{sat}}$ The axiom system of $\mathrm{BPA}^{\mathrm{sat}}$ consists of axioms A1-A5, A6ID and A7ID from Table 1 and the equations given in Table 4.

$\sigma^{0}_{\mathsf{abs}}(x) = \overline{\upsilon}^{0}_{\mathsf{abs}}(x)$	SAT1	$\sigma^{p}_{\mathsf{abs}}(\dot{\delta}) \cdot x = \sigma^{p}_{\mathsf{abs}}(\dot{\delta})$	SAT6
$\sigma^{p}_{\mathsf{abs}}(\sigma^{q}_{\mathsf{abs}}(x)) = \sigma^{p+q}_{\mathsf{abs}}(x)$	SAT2	$\tilde{a} + \tilde{\delta} = \tilde{a}$	A6SAa
$\sigma^{p}_{\mathsf{abs}}(x) + \sigma^{p}_{\mathsf{abs}}(y) = \sigma^{p}_{\mathsf{abs}}(x+y)$	SAT3	$\sigma^{r}_{\mathsf{abs}}(x) + \tilde{\delta} = \sigma^{r}_{\mathsf{abs}}(x)$	A6SAb
$\sigma^{p}_{\mathsf{abs}}(x) \cdot \upsilon^{p}_{\mathsf{abs}}(y) = \sigma^{p}_{\mathsf{abs}}(x \cdot \dot{\delta})$	SAT4	$\tilde{\delta} \cdot x = \tilde{\delta}$	A7SA
$\sigma^{p}_{\mathsf{abs}}(x) \cdot (\upsilon^{p}_{\mathsf{abs}}(y) + \sigma^{p}_{\mathsf{abs}}(z)) =$ $\sigma^{p}_{\mathsf{abs}}(x \cdot \overline{\upsilon}^{0}_{\mathsf{abs}}(z))$	SAT5		
		$\overline{\upsilon}^{0}_{\mathsf{abs}}(\dot{\delta}) = \dot{\delta}$	SAI0a
$\upsilon^{p}_{\mathsf{abs}}(\dot{\delta}) = \dot{\delta}$	SATO0	$\overline{\upsilon}^{r}_{\mathsf{abs}}(\dot{\delta}) = \sigma^{r}_{\mathsf{abs}}(\dot{\delta})$	SAI0b
$\upsilon^{0}_{\mathsf{abs}}(x) = \dot{\delta}$	SATO1	$\overline{\upsilon}^{0}_{\mathsf{abs}}(\tilde{a}) = \tilde{a}$	SAI1
$\upsilon^{r}_{\mathsf{abs}}(\tilde{a}) = \tilde{a}$	SATO2	$\overline{\upsilon}^{r}_{\mathsf{abs}}(\tilde{a}) = \sigma^{r}_{\mathsf{abs}}(\dot{\delta})$	SAI2
$\upsilon^{p+q}_{\mathsf{abs}}(\sigma^{p}_{\mathsf{abs}}(x)) = \sigma^{p}_{\mathsf{abs}}(\upsilon^{q}_{\mathsf{abs}}(x))$	SATO3	$\overline{\upsilon}^{p+q}_{\mathsf{abs}}(\sigma^{p}_{\mathsf{abs}}(x)) = \sigma^{p}_{\mathsf{abs}}(\overline{\upsilon}^{q}_{\mathsf{abs}}(\overline{\upsilon}^{0}_{\mathsf{abs}}(x)))$	SAI3
$\upsilon^{p}_{\mathsf{abs}}(x+y) = \upsilon^{p}_{\mathsf{abs}}(x) + \upsilon^{p}_{\mathsf{abs}}(y)$	SATO4	$\overline{\upsilon}^{p}_{\mathsf{abs}}(x+y) = \overline{\upsilon}^{p}_{\mathsf{abs}}(x) + \overline{\upsilon}^{p}_{\mathsf{abs}}(y)$	SAI4
$\upsilon^{p}_{\mathsf{abs}}(x \cdot y) = \upsilon^{p}_{\mathsf{abs}}(x) \cdot y$	SATO5	$\overline{\upsilon}^{p}_{\mathsf{abs}}(x \cdot y) = \overline{\upsilon}^{p}_{\mathsf{abs}}(x) \cdot y$	SAI5

Table 4: Axioms of $\mathrm{BPA}^{\mathrm{sat}}$ ($a \in \mathsf{A}_{\delta}$, $p, q \geq 0$, $r > 0$)

Axioms SAT1-SAT6, A6SAa, SATO0-SATO5 and SAI0-SAI5 are simple reformulations of axioms DAT1-DAT6, A6DAa, DATO0-DATO5 and DAI0-DAI5 of $\mathrm{BPA}^{\mathrm{dat}}$. Unlike their counterparts in $\mathrm{BPA}^{\mathrm{dat}}$, axioms A6SAb and A7SA are not derivable. The cause of this is the absence of a counterpart of axiom DAT7. Like in the case of $\mathrm{BPA}^{\mathrm{dat}}$, we have that for all closed $\mathrm{BPA}^{\mathrm{sat}}$-terms t and for all $p > 0$ either there exists a closed term t' such that $t = \upsilon^{p}_{\mathsf{abs}}(t')$ is derivable or there exist closed terms t' and t'' such that $t = \upsilon^{p}_{\mathsf{abs}}(t') + \sigma^{p}_{\mathsf{abs}}(t'')$ is derivable. Besides, $\overline{\upsilon}^{0}_{\mathsf{abs}}(t) = t$ is derivable for all closed $\mathrm{BPA}^{\mathrm{sat}}$-terms t.

3 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 $\mathrm{BPA}^{\mathrm{sat}}$. The integration operator $\int$ provides for alternative composition over a continuum of alternatives. The process $\int_{v \in V} t$, where v is a variable ranging over $\mathbb{R}^{+}$, $V \subseteq \mathbb{R}^{+}$ and t is a term that contains no other free variable than v, behaves like one of the alternatives $t[p/v]$ for $p \in V$.

The extension of $\mathrm{BPA}^{\mathrm{sat}}$ with the integration operator is called $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$.

Axioms of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ The axiom system of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ consists of the axioms of $\mathrm{BPA}^{\mathrm{sat}}$ and the equations given in Table 5. We use F and G as variables rang-

$\int_{v \in V} F(v) = \int_{w \in V} F(w)$	INT1	$V \neq \emptyset \Rightarrow \int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\dot{\delta}) = \sigma^{\sup V}_{\mathsf{abs}}(\dot{\delta})$	INT7SA
$\int_{v \in \emptyset} F(v) = \dot{\delta}$	INT2	$V \neq \emptyset,\ \sup V \notin V \Rightarrow \int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\tilde{\delta}) = \sigma^{\sup V}_{\mathsf{abs}}(\dot{\delta})$	INT8SA
$\int_{v \in \{p\}} F(v) = F(p)$	INT3	$\sup V \in V \Rightarrow \int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\tilde{\delta}) = \sigma^{\sup V}_{\mathsf{abs}}(\tilde{\delta})$	INT9SA
$\int_{v \in V \cup W} F(v) = \int_{v \in V} F(v) + \int_{v \in W} F(v)$	INT4	$\int_{v \in V} \sigma^{p}_{\mathsf{abs}}(F(v)) = \sigma^{p}_{\mathsf{abs}}(\int_{v \in V} F(v))$	INT10SA
$V \neq \emptyset \Rightarrow \int_{v \in V} x = x$	INT5		
$(\forall p \in V \bullet F(p) = G(p)) \Rightarrow \int_{v \in V} F(v) = \int_{v \in V} G(v)$	INT6		
$\int_{v \in V} (F(v) + G(v)) = \int_{v \in V} F(v) + \int_{v \in V} G(v)$	INT11	$\upsilon^{p}_{\mathsf{abs}}(\int_{v \in V} F(v)) = \int_{v \in V} \upsilon^{p}_{\mathsf{abs}}(F(v))$	SATO6
$\int_{v \in V} (F(v) \cdot x) = (\int_{v \in V} F(v)) \cdot x$	INT12	$\overline{\upsilon}^{p}_{\mathsf{abs}}(\int_{v \in V} F(v)) = \int_{v \in V} \overline{\upsilon}^{p}_{\mathsf{abs}}(F(v))$	SAI6

Table 5: Additional axioms for $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ ($p \geq 0$)

ing over functions from non-negative real numbers to processes with standard absolute timing – which are represented by terms that contain a designated free variable ranging over the non-negative real numbers.

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 an form of alternative composition over a set of alternatives. Axioms INT10SA, INT11, INT12, SATO6 and SAI6 can simply be regarded as variants of axioms SAT3, A2, A4, SATO4 and

SAI4, respectively. Notice that $\int_{v \in V} \sigma^v_{\mathsf{abs}}(\dot{\delta})$ and $\int_{v \in V} \sigma^v_{\mathsf{abs}}(\tilde{\delta})$ are indistinguishable if the supremum of V is not in V.

Example 3 We consider the polling process of Examples 1 and 2 again. It can be recursively defined in BPA$^{\mathrm{sat}}$I as follows:

$$\begin{aligned} \mathit{Poll} = & \int_{t \in [0,1)} \sigma^t_{\mathsf{abs}}(\textstyle\sum_{d \in D} \widetilde{r_1(d)} \cdot \widetilde{s_3(d)}) + \\ & \int_{t \in [1,2)} \sigma^t_{\mathsf{abs}}(\textstyle\sum_{d \in D} \widetilde{r_2(d)} \cdot \widetilde{s_3(d)}) + \sigma^2_{\mathsf{abs}}(\mathit{Poll}) \end{aligned}$$

Notice that, unlike in BPA$^{\mathrm{drt}}$ and BPA$^{\mathrm{dat}}$, we are able to describe that an accepted datum is immediately delivered.

4 Initial abstraction

In this subsection, we will informally introduce the concept of initial abstraction in the setting of BPA$^{\mathrm{dat}}$ and BPA$^{\mathrm{sat}}$. It is considered to be the primary way of forming processes with parametric timing. The ways of combining and timing processes available in BPA$^{\mathrm{dat}}$ and BPA$^{\mathrm{sat}}$ can simply be lifted to processes with parametric timing.

Initial abstraction is the basis for an alternative way to deal with relative timing. Because it builds upon process algebra with absolute timing, we can thus integrate absolute timing and relative timing.

The behaviour of processes with parametric timing depend on their initialization time. They can be perceived as functions from natural numbers to processes with discrete absolute timing that map each natural number n to a process initialized at time n in the discrete case and as functions from non-negative real numbers to processes with continuous absolute timing that map each non-negative real number p to a process initialized at time p in the continuous case. Initial abstraction is an abstraction mechanism to form such functions. It is reminiscent of λ-abstraction in the λ-calculus, but specific to the case where the parameter is process initialization time: the process $\surd_{\mathsf{d}} i \,.\, t$, where i is a variable ranging over $\mathbb{N}$ and t is a term that contains no other variable than i, behaves like $t[n/i]$ when initialized at time n; and the process $\surd_{\mathsf{s}} v \,.\, t$, where v is a variable ranging over $\mathbb{R}^+$ and t is a term that contains no other variable than v, behaves like $t[p/v]$ when initialized at time p.

First, we extend BPA$^{\mathrm{dat}}$ with the discrete initial abstraction operator $\surd_{\mathsf{d}}$. The resulting theory is called BPA$^{\mathrm{dat}}\surd$.

Axioms of BPA$^{\mathrm{dat}}\surd$ The axiom system of BPA$^{\mathrm{dat}}\surd$ consists of the axioms of BPA$^{\mathrm{dat}}$ and the equations given in Table 6. We use F and G like before in Table 5.

Axioms DIA1 and DIA2 are similar to the α- and β-conversion rules of λ-calculus. Axiom DIA3 shows that multiple discrete initial abstractions can be replaced by one. Axiom DIA4 points out that processes with discrete absolute timing are special cases of processes with discrete parametric timing: they simply do

$\surd_{\mathsf{d}} i \,.\, F(i) = \surd_{\mathsf{d}} j \,.\, F(j)$	DIA1	$\sigma^n_{\mathsf{abs}}(\underline{a}) \cdot x = \sigma^n_{\mathsf{abs}}(\underline{a}) \cdot \overline{\upsilon}^n_{\mathsf{abs}}(x)$	DIA6
$\overline{\upsilon}^n_{\mathsf{abs}}(\surd_{\mathsf{d}} i \,.\, F(i)) = \overline{\upsilon}^n_{\mathsf{abs}}(F(n))$	DIA2	$\sigma^n_{\mathsf{abs}}(\surd_{\mathsf{d}} i \,.\, F(i)) = \sigma^n_{\mathsf{abs}}(F(0))$	DIA7
$\surd_{\mathsf{d}} i \,.\, (\surd_{\mathsf{d}} j \,.\, K(i,j)) = \surd_{\mathsf{d}} i \,.\, K(i,i)$	DIA3	$(\surd_{\mathsf{d}} i \,.\, F(i)) + x = \surd_{\mathsf{d}} i \,.\, (F(i) + \overline{\upsilon}^i_{\mathsf{abs}}(x))$	DIA8
$x = \surd_{\mathsf{d}} i \,.\, x$	DIA4	$(\surd_{\mathsf{d}} i \,.\, F(i)) \cdot x = \surd_{\mathsf{d}} i \,.\, (F(i) \cdot x)$	DIA9
$(\forall n \in \mathbb{N} \bullet \overline{\upsilon}^n_{\mathsf{abs}}(x) = \overline{\upsilon}^n_{\mathsf{abs}}(y)) \Rightarrow x = y$	DIA5	$\upsilon^n_{\mathsf{abs}}(\surd_{\mathsf{d}} i \,.\, F(i)) = \surd_{\mathsf{d}} i \,.\, \upsilon^n_{\mathsf{abs}}(F(i))$	DIA10

Table 6: Axioms for discrete initial abstraction ($n \geq 0$)

not vary with different initialization times. Axiom DIA5 is an extensionality axiom. Axiom DIA6 expresses that in case a process performs an action and then proceeds as another process, the initialization time of the latter process is the time at which the action is performed. The remaining axioms concern the lifting of the ways of combining and timing processes available in $\mathrm{BPA}^{\mathrm{dat}}$ to processes with discrete parametric timing.

Next, we extend $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ with the standard initial abstraction operator $\surd_{\mathsf{s}}$. The resulting theory is called $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$.

Axioms of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ The axiom system of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ consists of the axioms of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ and the equations given in Table 7. We use F and G just as before

$\surd_{\mathsf{s}} v \,.\, F(v) = \surd_{\mathsf{s}} w \,.\, F(w)$	SIA1	$\sigma^p_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \sigma^p_{\mathsf{abs}}(F(0))$	SIA7
$\overline{\upsilon}^p_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \overline{\upsilon}^p_{\mathsf{abs}}(F(p))$	SIA2	$(\surd_{\mathsf{s}} v \,.\, F(v)) + x = \surd_{\mathsf{s}} v \,.\, (F(v) + \overline{\upsilon}^v_{\mathsf{abs}}(x))$	SIA8
$\surd_{\mathsf{s}} v \,.\, (\surd_{\mathsf{s}} w \,.\, K(v,w)) = \surd_{\mathsf{s}} v \,.\, K(v,v)$	SIA3	$(\surd_{\mathsf{s}} v \,.\, F(v)) \cdot x = \surd_{\mathsf{s}} v \,.\, (F(v) \cdot x)$	SIA9
$x = \surd_{\mathsf{s}} v \,.\, x$	SIA4	$\upsilon^p_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, \upsilon^p_{\mathsf{abs}}(F(v))$	SIA10
$(\forall p \in \mathbb{R}^+ \bullet \overline{\upsilon}^p_{\mathsf{abs}}(x) = \overline{\upsilon}^p_{\mathsf{abs}}(y)) \Rightarrow x = y$	SIA5	$\int_{v \in V} (\surd_{\mathsf{s}} w \,.\, K(v,w)) = \surd_{\mathsf{s}} w \,.\, (\int_{v \in V} K(v,w))$ if $v \neq w$	SIA17
$\sigma^p_{\mathsf{abs}}(\tilde{a}) \cdot x = \sigma^p_{\mathsf{abs}}(\tilde{a}) \cdot \overline{\upsilon}^p_{\mathsf{abs}}(x)$	SIA6		

Table 7: Axioms for standard initial abstraction ($p \geq 0$)

in Table 5.

Except for axiom SIA17, the axioms for standard initial abstraction are simple reformulations of axioms for discrete initial abstraction. Axiom SIA17 concern the lifting of integration, which is not available in $\mathrm{BPA}^{\mathrm{dat}}\surd$, to processes with standard parametric timing.

5 Integration revisited

We now return to integration. It will be added to $\mathrm{BPA}^{\mathrm{srt}}$ as well. The rules for the operational semantics concerning integration will be given both for the case of absolute timing and the case of relative timing. This will show that the rules for the operational semantics concerning integration are simple in case of absolute timing, but complex in case of relative timing. The origin of this complexity turns out to be that, unlike in case of absolute timing, in case of relative timing a process always changes into another process while idling.

First, we extend BPA$^{\text{srt}}$ with the integration operator $\int$. The resulting theory is called BPA$^{\text{srt}}$I.

AXIOMS OF BPA$^{\text{srt}}$I The axiom system of BPA$^{\text{srt}}$I consists of the axioms of BPA$^{\text{srt}}$, axioms INT1-INT6, INT11 and INT12 from Table 5 and the equations given in Table 8.

$V \neq \emptyset \Rightarrow \int_{v \in V} \sigma^{v}_{\mathsf{rel}}(\dot{\delta}) = \sigma^{\sup V}_{\mathsf{rel}}(\dot{\delta})$	INT7SR	$\int_{v \in V} \sigma^{p}_{\mathsf{rel}}(F(v)) = \sigma^{p}_{\mathsf{rel}}(\int_{v \in V} F(v))$	INT10SR
$V \neq \emptyset,\ \sup V \notin V \Rightarrow$			
$\int_{v \in V} \sigma^{v}_{\mathsf{rel}}(\tilde{\tilde{\delta}}) = \sigma^{\sup V}_{\mathsf{rel}}(\dot{\delta})$	INT8SR		
$\sup V \in V \Rightarrow$		$\upsilon^{p}_{\mathsf{rel}}(\int_{v \in V} F(v)) = \int_{v \in V} \upsilon^{p}_{\mathsf{rel}}(F(v))$	SRTO6
$\int_{v \in V} \sigma^{v}_{\mathsf{rel}}(\tilde{\tilde{\delta}}) = \sigma^{\sup V}_{\mathsf{rel}}(\tilde{\tilde{\delta}})$	INT9SR	$\overline{\upsilon}^{p}_{\mathsf{rel}}(\int_{v \in V} F(v)) = \int_{v \in V} \overline{\upsilon}^{p}_{\mathsf{rel}}(F(v))$	SRI6

Table 8: Additional axioms for BPA$^{\text{srt}}$I ($p \geq 0$)

Axioms INT7SR-INT10SR, SRTO6 and SRI6 are trivial reformulations of axioms INT7SA-INT10SA, SATO6 and SAI6 of BPA$^{\text{sat}}$I. In other words, the axioms concerning integration in case of relative timing are essentially the same as the ones in case of absolute timing.

EXAMPLE 4 We consider the polling process of Examples 1, 2 and 3 again. It can be recursively defined in BPA$^{\text{srt}}$I as follows:

$$\begin{aligned} \mathit{Poll} = {} & \int_{t \in [0,1)} \sigma^{t}_{\mathsf{rel}}(\textstyle\sum_{d \in D} \widetilde{\widetilde{r_1(d)}} \cdot \widetilde{\widetilde{s_3(d)}}) + {} \\ & \int_{t \in [1,2)} \sigma^{t}_{\mathsf{rel}}(\textstyle\sum_{d \in D} \widetilde{\widetilde{r_2(d)}} \cdot \widetilde{\widetilde{s_3(d)}}) + \sigma^{2}_{\mathsf{rel}}(\mathit{Poll}) \end{aligned}$$

If the process defined above is started up at time 0, this process behaves exactly the same as the process *Poll* from Example 3.

Next, we give the rules for the operational semantics of integration in case of relative timing.

SEMANTICS OF BPA$^{\text{srt}}$I The structural operational semantics of BPA$^{\text{srt}}$I is described by the rules for BPA$^{\text{srt}}$ and the rules given in Table 9. As for BPA$^{\text{srt}}$,

$$\frac{F(q) \xrightarrow{a} x'}{\int_{v \in V} F(v) \xrightarrow{a} x'}\ q \in V \qquad \frac{F(q) \xrightarrow{a} \surd}{\int_{v \in V} F(v) \xrightarrow{a} \surd}\ q \in V$$

$$\frac{\{F(q) \overset{r}{\longmapsto} F_1(q) \mid q \in V_1\},\ \ldots,\ \{F(q) \overset{r}{\longmapsto} F_n(q) \mid q \in V_n\},\quad \{F(q) \overset{r}{\not\longmapsto} \mid q \in V_{n+1}\}}{\int_{v \in V} F(v) \overset{r}{\longmapsto} \int_{v \in V_1} F_1(v) + \ldots + \int_{v \in V_n} F_n(v)}\ \begin{array}{l}\{V_1, \ldots, V_n\} \text{ partition of } V - V_{n+1}, \\ V_{n+1} \subset V\end{array}$$

$$\frac{\{F(q)\uparrow \mid q \in V\}}{\int_{v \in V} F(v)\uparrow}$$

Table 9: Additional rules for BPA$^{\text{srt}}$I ($a \in \mathsf{A}$, $p, q \geq 0$, $r > 0$)

the following transition predicates are used in Table 9: a binary predicate $\_ \xrightarrow{a} \_$

and a unary predicate $\_ \xrightarrow{a} \surd$ for each $a \in \mathsf{A}$, a binary predicate $\_ \xmapsto{r} \_$ for each $r \in \mathbb{R}^{+}$ such that $r > 0$, and a unary predicate $\_\uparrow$. The transition predicates can be explained as follows:

- $t \xrightarrow{a} t'$: process t is capable of first performing action a at the current point of time and then proceeding as process t';
- $t \xrightarrow{a} \surd$: process t is capable of first performing action a at the current point of time and then terminating successfully;
- $t \xmapsto{r} t'$: process t is capable of first idling for a period of time r and then proceeding as process t';
- $t\uparrow$: process t has deadlocked before the current point of time.

We write $t \not\xmapsto{r}$ for the set of all transition formulas $\neg(t \xmapsto{r} t')$ where t' is a closed term of $\mathrm{BPA^{srt}}$.

In case of relative timing, a process changes into another process while idling. The complexity of the third rule for integration is caused by the fact that the processes $F(p)$ with $p \in V$ that are capable of idling need not change uniformly while idling. This is illustrated in Example 5 below. The non-uniformity is in all cases of a finite nature: the operational semantics gives for each operator at most three ways in which the different processes obtained by means of the operator may change while idling for a certain period of time. Hence, V can always be partitioned into a finite number of sets $V_1, \ldots, V_{n+1}$ (where V_{n+1} may be empty) such that for each $V' \in \{V_1, \ldots, V_n\}$ the processes $F(p)$ with $p \in V'$ change uniformly while idling, and the processes $F(p)$ with $p \in V_{n+1}$ are not capable of idling.

Example 5 We illustrate that a process with relative timing need not change uniformly while idling by showing how the process $\int_{v \in [0.6,1.8)} (\sigma_{\mathsf{rel}}^{v}(\tilde{\tilde{a}}) + \sigma_{\mathsf{rel}}^{v+1.5}(\tilde{\tilde{b}}))$ has changed after 1.2 time units:

$$\int_{v \in [0.6,1.8)} (\sigma_{\mathsf{rel}}^{v}(\tilde{\tilde{a}}) + \sigma_{\mathsf{rel}}^{v+1.5}(\tilde{\tilde{b}})) \xmapsto{1.2}$$
$$\int_{v \in [0.6,1.2)} \sigma_{\mathsf{rel}}^{v+0.3}(\tilde{\tilde{b}}) + \int_{v \in [1.2,1.2]} (\tilde{\tilde{a}} + \sigma_{\mathsf{rel}}^{v+0.3}(\tilde{\tilde{b}})) + \int_{v \in (1.2,1.8)} (\sigma_{\mathsf{rel}}^{v-1.2}(\tilde{\tilde{a}}) + \sigma_{\mathsf{rel}}^{v+0.3}(\tilde{\tilde{b}})).$$

Finally, we give the rules for the operational semantics of integration in case of absolute timing. It turns out that the rules for integration in case of absolute timing are much simpler than in case of relative timing.

Semantics of $\mathrm{BPA^{sat}I}$ The structural operational semantics of $\mathrm{BPA^{sat}I}$ is described by the rules for $\mathrm{BPA^{sat}}$ and the rules given in Table 10. As for $\mathrm{BPA^{sat}}$, see [6], the following transition predicates are used in Table 10: a binary predicate $\langle \_, p\rangle \xrightarrow{a} \langle \_, p\rangle$ and a unary predicate $\langle \_, p\rangle \xrightarrow{a} \langle \surd, p\rangle$ for each $a \in \mathsf{A}$ and $p \in \mathbb{R}^{+}$, a binary predicate $\langle \_, p\rangle \xmapsto{r} \langle \_, q\rangle$ for each $p, q, r \in \mathbb{R}^{+}$ such that $p + r = q$ and $r > 0$, and a unary predicate $\langle \_, p\rangle\uparrow$ for each $p \in \mathbb{R}^{+}$. The transition predicates can be explained as follows:

$$\frac{\langle F(q),p\rangle \xrightarrow{a} \langle x',p\rangle}{\langle \int_{v\in V} F(v),p\rangle \xrightarrow{a} \langle x',p\rangle}\, q\in V \quad \frac{\langle F(q),p\rangle \xrightarrow{a} \langle \surd,p\rangle}{\langle \int_{v\in V} F(v),p\rangle \xrightarrow{a} \langle \surd,p\rangle}\, q\in V$$

$$\frac{\langle F(q),p\rangle \overset{r}{\mapsto} \langle F(q),p+r\rangle}{\langle \int_{v\in V} F(v),p\rangle \overset{r}{\mapsto} \langle \int_{v\in V} F(v),p+r\rangle}\, q\in V \quad \frac{\{\langle F(q),p\rangle\uparrow \mid q\in V\}}{\langle \int_{v\in V} F(v),p\rangle\uparrow}$$

Table 10: Additional rules for $\mathrm{BPA^{sat}I}$ ($a \in \mathsf{A}$, $p,q \geq 0$, $r > 0$)

$\langle t,p\rangle \xrightarrow{a} \langle t',p\rangle$: process t is capable of first performing action a at point of time p and then proceeding as process t';

$\langle t,p\rangle \xrightarrow{a} \langle \surd,p\rangle$: process t is capable of first performing action a at point of time p and then terminating successfully;

$\langle t,p\rangle \overset{r}{\mapsto} \langle t,q\rangle$: process t is capable of first idling from point of time p to point of time q and then proceeding as process t;

$\langle t,p\rangle\uparrow$: process t has deadlocked before point of time p.

Here, it is worth remarking that the transition rules for $\mathrm{BPA^{sat}I}$ only define transition relations for which $\langle t,p\rangle \xrightarrow{a} \langle t',q\rangle$ and $\langle t,p\rangle \xrightarrow{a} \langle \surd,q\rangle$ never hold if $p \neq q$; and $\langle t,p\rangle \overset{r}{\mapsto} \langle t',q\rangle$ never holds if $t \not\equiv t'$.

6 Time-dependent state operator

We now turn to a time-dependent version of the state operator from [9]. The time-dependent state operator will be added to $\mathrm{BPA^{sat}I}$. It will become evident from this addition that, in case of relative timing, a more detailed operational semantics than the usual one is needed to deal with this operator. Thus, a possible advantage of relative timing, viz. an intuitively clearer operational semantics vanishes.

The state operator makes it easy to represent the execution of a process in a state. The basic idea is that the execution of an action in a state has effect on the state, i.e. it causes a change of state. Besides, there is an action left when an action is executed in a state. For example, in case the states are queues of data, when the action of instructing the addition or removal of a certain datum is executed in a state, the action of adding or removing that datum is left. The operator introduced here generalizes the state operator added to ACP without timing in [9]. The main difference with that operator is that the results of executing an action in a state may depend on time.

It is assumed that a fixed but arbitrary set S of *states* has been given, together with functions $\mathrm{act} : \mathsf{A} \times \mathbb{R}^{+} \times S \to \mathsf{A}_\delta$ and $\mathrm{eff} : \mathsf{A} \times \mathbb{R}^{+} \times S \to S$.

The state operator λ_s ($s \in S$) allows, given these functions, processes to interact with a state. Let p be a process. Then $\lambda_s(p)$ is the process p executed in state s. The function act gives, for each action a, time t and state s, the action that results from executing a in state s at time t. The function eff gives, for each action a, time t and state s, the state that results from executing a in state s at time t. The functions act and eff are extended to A_δ such that $\mathrm{act}(\delta,t,s) = \delta$ and $\mathrm{eff}(\delta,t,s) = s$ for all $t \in \mathbb{R}^{+}$ and $s \in S$.

First, we extend $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ with the time-dependent state operators λ_s.

Axioms for state operator The additional axioms for the *state* operators λ_s (for each $s \in S$) are given in Table 11.

$\lambda_s(\sigma^p_{\mathsf{abs}}(\dot{\delta})) = \sigma^p_{\mathsf{abs}}(\dot{\delta})$	SATSO0
$\lambda_s(\sigma^p_{\mathsf{abs}}(\tilde{a})) = \sigma^p_{\mathsf{abs}}(\mathrm{act}\widetilde{(a,p,s)})$	SATSO1
$\lambda_s(\sigma^p_{\mathsf{abs}}(\tilde{a} \cdot x)) = \sigma^p_{\mathsf{abs}}(\mathrm{act}\widetilde{(a,p,s)}) \cdot \lambda_{\mathrm{eff}(a,p,s)}(\sigma^p_{\mathsf{abs}}(x))$	SATSO2
$\lambda_s(x + y) = \lambda_s(x) + \lambda_s(y)$	SATSO3
$\lambda_s(\int_{v \in V} F(v)) = \int_{v \in V} \lambda_s(F(v))$	SATSO4

Table 11: Axioms for state operator ($a \in \mathsf{A}_\delta$, $p \geq 0$, $s \in S$)

These axioms reflect the intended meaning of the state operator clearly. They are reformulations of the axioms for the state operator added to ACP without timing in [9] which reflect the possible dependence on time.

Next, we give the rules for the operational semantics of the time-dependent state operator in case of absolute timing.

Semantics for state operator The structural operational semantics of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ extended with the state operator is described by the rules for $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}$ and the rules given in Table 12.

$$\frac{\langle x, p\rangle \xrightarrow{a} \langle x', p\rangle}{\langle \lambda_s(x), p\rangle \xrightarrow{\mathrm{act}(a,p,s)} \langle \lambda_{\mathrm{eff}(a,p,s)}(x'), p\rangle}\ \mathrm{act}(a,p,s) \neq \delta$$

$$\frac{\langle x, p\rangle \xrightarrow{a} \langle \surd, p\rangle}{\langle \lambda_s(x), p\rangle \xrightarrow{\mathrm{act}(a,p,s)} \langle \surd, p\rangle}\ \mathrm{act}(a,p,s) \neq \delta$$

$$\frac{\langle x, p\rangle \overset{r}{\mapsto} \langle x, p + r\rangle}{\langle \lambda_s(x), p\rangle \overset{r}{\mapsto} \langle \lambda_s(x), p + r\rangle} \qquad \frac{\langle x, p\rangle \uparrow}{\langle \lambda_s(x), p\rangle \uparrow}$$

Table 12: Rules for state operator ($a \in \mathsf{A}$, $p \geq 0$, $r > 0$, $s \in S$)

In case of absolute timing, the operational semantics gives the capabilities of processes related to points of time (see Section 5). Therefore, the operational semantics is detailed enough to deal with the time-dependent state operator, which requires that the points of time at which actions are performed are available. However, in case of relative timing, the usual operational semantics is less detailed: all capabilities are implicitly at the current point of time (see Section 5), which may be any point of time. Neither the points of time at which actions are performed nor the periods of time passed since previous actions were performed are available. Therefore, the usual operational semantics is not detailed enough to deal with the time-dependent state operator. In order to add this operator to $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$, a more detailed operational semantics of $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ is needed with transition predicates that are virtually the same as the ones used for $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$.

The operational semantics would be much like the operational semantics of the version of ACP with continuous relative timing described in [1]. The rules for the state operator would be almost the same as the ones given above. The only difference is that the second occurrence of the variable x in the premise and the conclusion of the third rule must be a fresh variable x'. It is clear that a possible advantage of relative timing, viz. an intuitively clearer operational semantics, vanishes.

Of course, there is always the alternative to extend $\mathrm{BPA^{srt}I}$ with a different state operator with which the original one can be mimiced rather directly: one for which the results of executing an action in a state do not depend on time, but for which idling may cause a change of state. We can also add a time-dependent version of the process creation operator from [9] to $\mathrm{BPA^{srt}I}$ and $\mathrm{BPA^{sat}I}$. Again, the addition to $\mathrm{BPA^{srt}I}$ requires the more detailed operational semantics of $\mathrm{BPA^{srt}I}$ mentioned above. Unlike in the case of the state operator, there is not the alternative to extend $\mathrm{BPA^{srt}I}$ with a different process creation operator with which the original one can be mimiced.

7 Embeddings

It is interesting to know how the different process algebras introduced in this paper are related to each other. We establish formal connections in the form of embeddings. An embedding is a term structure preserving injective mapping from the terms of one process algebra to the terms of another process algebra such that what is derivable for closed terms in the former process algebra remains derivable after mapping in the latter process algebra. As usual, we characterize each embedding by explicit definitions of the constants and operators of the source process algebra that are not available in the target process algebra (see [6] for details). Perhaps rather unexpectedly, there does not exist an embedding of $\mathrm{BPA^{drt}}$ in $\mathrm{BPA^{srt}I}$.

The following constants and operators of $\mathrm{BPA^{drt}}$ are not present in $\mathrm{BPA^{dat}}\surd$: $\underline{\underline{a}}$ $(a \in \mathsf{A}_\delta)$, σ_{rel}, υ_{rel} and $\overline{\upsilon}_{\mathsf{rel}}$. Explicit definitions of these constants and operators in $\mathrm{BPA^{dat}}\surd$ are given in Table 13. These definitions induce an embedding of

$\underline{\underline{a}} = \surd_{\mathsf{d}} i \cdot \sigma^{i}_{\mathsf{abs}}(\underline{a})$ for each $a \in \mathsf{A}_\delta$
$\sigma^{n}_{\mathsf{rel}}(x) = \surd_{\mathsf{d}} i \cdot \overline{\upsilon}^{i+n}_{\mathsf{abs}}(x)$
$\upsilon^{n}_{\mathsf{rel}}(x) = \surd_{\mathsf{d}} i \cdot \upsilon^{i+n}_{\mathsf{abs}}(\overline{\upsilon}^{i}_{\mathsf{abs}}(x))$
$\overline{\upsilon}^{n}_{\mathsf{rel}}(x) = \surd_{\mathsf{d}} i \cdot \overline{\upsilon}^{i+n}_{\mathsf{abs}}(\overline{\upsilon}^{i}_{\mathsf{abs}}(x))$

Table 13: Explicit definition constants/operators of $\mathrm{BPA^{drt}}$ in $\mathrm{BPA^{dat}}\surd$

$\mathrm{BPA^{drt}}$ in $\mathrm{BPA^{dat}}\surd$. The proof is essentially the same as the proof of Theorem 6 in [6]. The embedding demonstrates that there is no reason to distinguish the deadlocked process in case of absolute timing from the deadlocked process in case of relative timing. This can be explained in $\mathrm{BPA^{dat}}\surd$ by the derivability of $\dot{\delta} = \surd_{\mathsf{d}} i \cdot \sigma^{i}_{\mathsf{abs}}(\dot{\delta})$, which looks like a definition of the deadlocked process of $\mathrm{BPA^{drt}}$ in $\mathrm{BPA^{dat}}\surd$.

The following constants and operators of $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ are not present in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$: $\tilde{\tilde{a}}$ ($a \in \mathsf{A}_\delta$), σ_{rel}, υ_{rel} and $\overline{\upsilon}_{\mathsf{rel}}$. Explicit definitions of these constants and operators in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ are given in Table 14. These definitions induce an embedding of

$$
\begin{array}{l}
\tilde{\tilde{a}} = \surd_{\mathsf{s}} v \,.\, \sigma^{v}_{\mathsf{abs}}(\tilde{a}) \quad \text{for each } a \in \mathsf{A}_\delta \\
\sigma^{p}_{\mathsf{rel}}(x) = \surd_{\mathsf{s}} v \,.\, \overline{\upsilon}^{v+p}_{\mathsf{abs}}(x) \\
\upsilon^{p}_{\mathsf{rel}}(x) = \surd_{\mathsf{s}} v \,.\, \upsilon^{v+p}_{\mathsf{abs}}(\overline{\upsilon}^{v}_{\mathsf{abs}}(x)) \\
\overline{\upsilon}^{p}_{\mathsf{rel}}(x) = \surd_{\mathsf{s}} v \,.\, \overline{\upsilon}^{v+p}_{\mathsf{abs}}(\overline{\upsilon}^{v}_{\mathsf{abs}}(x))
\end{array}
$$

Table 14: Explicit definition constants/operators of $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$

$\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$. The proof is given in [6] (Theorem 6).

The two above-mentioned embeddings correspond to the view that, for a process with relative timing, the execution of its first action is always timed relative to the initialization time of the process.

The following constants and operators of $\mathrm{BPA}^{\mathrm{dat}}\surd$ are absent in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$: $\underline{a}$ ($a \in \mathsf{A}_\delta$) and $\surd_{\mathsf{d}}$. Besides, the operators σ_{abs}, υ_{abs} and $\overline{\upsilon}_{\mathsf{abs}}$ have a natural number instead of a non-negative real number as their first argument. Explicit definitions of these constants and operators in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ are given in Table 15.

$$
\begin{array}{l}
\underline{a} = \int_{v \in [0,1)} \sigma^{v}_{\mathsf{abs}}(\tilde{a}) \quad \text{for each } a \in \mathsf{A}_\delta \\
\sigma^{n}_{\mathsf{abs}}(x) = \sigma^{n}_{\mathsf{abs}}(x) \\
\upsilon^{n}_{\mathsf{abs}}(x) = \upsilon^{n}_{\mathsf{abs}}(x) \\
\overline{\upsilon}^{n}_{\mathsf{abs}}(x) = \overline{\upsilon}^{n}_{\mathsf{abs}}(x) \\
\surd_{\mathsf{d}} i \,.\, F(i) = \surd_{\mathsf{s}} v \,.\, F(\lfloor v \rfloor)
\end{array}
$$

Table 15: Explicit definition constants/operators of $\mathrm{BPA}^{\mathrm{dat}}\surd$ in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$

Notice that the explicit definitions of the operators σ_{abs}, υ_{abs} and $\overline{\upsilon}_{\mathsf{abs}}$ express that they are the restrictions of the corresponding operators of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ to $\mathbb{N}$. The definitions given in Table 15 induce an embedding of $\mathrm{BPA}^{\mathrm{dat}}\surd$ in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$. The proof is given in [6] (Theorem 12).

The embedding concerned corresponds to the view that, for a discrete time process, the execution of its first action is always timed with respect to a time interval with discrete bounds (left closed, right open).

In summary, there exists an embedding of $\mathrm{BPA}^{\mathrm{drt}}$ in $\mathrm{BPA}^{\mathrm{dat}}\surd$, an embedding of $\mathrm{BPA}^{\mathrm{dat}}\surd$ in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ and an embedding of $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$. Thus, we have established a formal connection between $\mathrm{BPA}^{\mathrm{drt}}$ and $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$.

However, there does not exist an embedding of $\mathrm{BPA}^{\mathrm{drt}}$ in $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$. The term structure preservation required for an embedding fails due to the lack of a general mechanism for parametric timing in $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$. The heart of the problem is that we cannot produce explicit definitions in $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ for the constants $\underline{\underline{a}}$ ($a \in \mathsf{A}_\delta$) of $\mathrm{BPA}^{\mathrm{drt}}$. The plausible definition $\underline{\underline{a}} = \int_{v \in [0,1)} \sigma^{v}_{\mathsf{rel}}(\tilde{\tilde{a}})$ for each $a \in \mathsf{A}_\delta$ is incorrect. According to these definitions, the process $\underline{\underline{a}} \cdot \underline{\underline{b}}$ does not have to perform both $\underline{\underline{a}}$ and $\underline{\underline{b}}$ in the current time slice: $\underline{\underline{b}}$ may also be performed in the first next time slice. The point is that the maximal relative delay of $\underline{\underline{b}}$ should be less than

1 depending on what $\underline{\underline{a}}$ has left over. To express that dependency, we need a mechanism for parametric timing like initial abstraction.

This suggests that a discretization operator cannot be added to $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$. By the lack of an embedding, it is not even clear what processes in the model of $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ are to be considered discretized.

8 Discretization

Consider the subset of processes in the model of $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ generated by the embedded constants and operators of $\mathrm{BPA}^{\mathrm{dat}}\surd$. This set can be characterized as the set of those processes with standard parametric timing that are discretized.

We define the notion of a discretized process with standard parametric timing in terms of the *discretization* operator $\mathcal{D}$ of which the defining axioms are given in Table 16. The transition rules for discretization on processes with standard absolute timing are given in Table 17. These rules show that discretization

$$\begin{array}{l}
\mathcal{D}(\dot{\delta}) = \dot{\delta} \\
\mathcal{D}(\tilde{a}) = \int_{v \in [0,1)} \sigma^{v}_{\mathsf{abs}}(\tilde{a}) \\
\mathcal{D}(\sigma^{p}_{\mathsf{abs}}(x)) = \sigma^{\lfloor p \rfloor}_{\mathsf{abs}}(\mathcal{D}(x)) \\
\mathcal{D}(x + y) = \mathcal{D}(x) + \mathcal{D}(y) \\
\mathcal{D}(x \cdot y) = \mathcal{D}(x) \cdot \mathcal{D}(y) \\
\mathcal{D}(\int_{v \in V} F(v)) = \int_{v \in V} \mathcal{D}(F(v)) \\
\mathcal{D}(\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, \mathcal{D}(F(v))
\end{array}$$

Table 16: Axioms for discretization ($a \in \mathsf{A}_{\delta}$, $p \geq 0$)

$$\frac{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}{\langle \mathcal{D}(x), q \rangle \xrightarrow{a} \langle \mathcal{D}(x'), q \rangle}\, q \in [\lfloor p \rfloor, \lfloor p \rfloor + 1) \qquad \frac{\langle x, p \rangle \xrightarrow{a} \langle \surd, p \rangle}{\langle \mathcal{D}(x), q \rangle \xrightarrow{a} \langle \surd, q \rangle}\, q \in [\lfloor p \rfloor, \lfloor p \rfloor + 1)$$

$$\frac{\langle x, p \rangle \overset{r}{\longmapsto} \langle x', p + r \rangle}{\langle \mathcal{D}(x), p \rangle \overset{r'}{\longmapsto} \langle \mathcal{D}(x), p + r' \rangle}\, p + r' \in [p + r, \lfloor p + r \rfloor + 1) \qquad \frac{\langle x, p \rangle \uparrow}{\langle \mathcal{D}(x), p \rangle \uparrow}$$

$$\frac{\langle x, p \rangle \not\uparrow}{\langle \mathcal{D}(x), p \rangle \overset{r}{\longmapsto} \langle \mathcal{D}(x), p + r \rangle}\, p + r \in (p, \lfloor p \rfloor + 1)$$

Table 17: Rules for discretization ($a \in \mathsf{A}$, $r, r' > 0$)

extends the capabilities of a process at any point of time to the whole time slice in which the point of time occurs. Discretization can simply be lifted to processes with standard parametric timing, viz. pointwise.

A process with standard parametric timing x is *discretized* if $x = \mathcal{D}(x)$. For any such process, the following holds: if an action can be performed at some time p, it can also be performed at any other time p' such that $\lfloor p \rfloor \leq p' < \lfloor p \rfloor + 1$.

The set of discretized processes with standard parametric timing is closed under the embedded operators of $\mathrm{BPA}^{\mathrm{dat}}\surd$. It is even the smallest such set that includes the embedded constants of $\mathrm{BPA}^{\mathrm{dat}}\surd$. This suggests the construction of a model of $\mathrm{BPA}^{\mathrm{dat}}\surd$. That model happens to be isomorphic to the standard model of $\mathrm{BPA}^{\mathrm{dat}}\surd$. In point of fact, the discretization operator turns the processes that are considered in $\mathrm{BPA}^{\mathrm{sat}}\mathrm{I}\surd$ into the processes that are considered in $\mathrm{BPA}^{\mathrm{dat}}\surd$.

We introduced the discretization operator in order to define the notion of a discretized process. However, that is not the only application of this operator. Having a closed term t denoting some process with standard absolute timing, apposite change of the time scale may yield a closed term t' such that $t' = \mathcal{D}(t')$. A change of the time scale is apposite if the process can faithfully be considered at the more abstract level where time is measured on a discrete time scale. The point here is that the abstraction obtained by the discretization makes the process better amenable to analysis.

Example 6 The polling process defined in Example 2 behaves exactly the same as the discretization of the polling process defined in Example 3. After discretization, immediate delivery of data becomes delivery while the input port is still enabled. This was to be expected because the time unit used is the time that an input port is enabled without a pause.

Unfortunately, the discretization operator cannot be added to $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$. The problem of adding discretization to $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ is closely related to the problem of finding an embedding of $\mathrm{BPA}^{\mathrm{drt}}$ in $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$. We cannot discretize the constants $\tilde{\tilde{a}}$ ($a \in \mathsf{A}_\delta$) of $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ in a satisfactory way. The plausible axiom $\mathcal{D}(\tilde{\tilde{a}}) = \int_{v\in[0,1)} \sigma^v_{\mathsf{rel}}(\tilde{a})$ for each $a \in \mathsf{A}_\delta$ is incorrect. According to these axioms, the process $\mathcal{D}(\tilde{\tilde{a}}) \cdot \mathcal{D}(\tilde{\tilde{b}})$ does not have to perform both $\mathcal{D}(\tilde{\tilde{a}})$ and $\mathcal{D}(\tilde{\tilde{b}})$ in the current time slice: $\mathcal{D}(\tilde{\tilde{b}})$ may also be performed in the first next time slice. Obviously, like in case of the embedding of $\mathrm{BPA}^{\mathrm{drt}}$ in $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$, discretization cannot be added to $\mathrm{BPA}^{\mathrm{srt}}\mathrm{I}$ without first adding a mechanism for parametric timing.

9 Concluding remarks

I have presented some point that are peculiar to relative timing. These points shed a new light on the merits of relative timing. As far as I know, they have not been presented before.

The intricacy of the rules for the operational semantics concerning integration in case of relative timing, as demonstrated in Section 5, remained unnoticed in [1] because wrong rules were given there. In [6], integration is only presented for the case of absolute timing. A state operator for which idling may cause a change of state, as mentioned at the end of Section 6, is presented in [11] for the case that the time scale is discrete.

The rules for the operational semantics concerning integration in case of relative timing (Section 5), as well as the rules for the operational semantics concerning discretization in case of absolute timing (Section 8), have not been presented before.

References

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

[2] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra (extended abstract). In W.R. Cleaveland, editor, *CONCUR'92*, pages 401–420. LNCS 630, Springer-Verlag, 1992. Full version: Report P9208b, Programming Research Group, University of Amsterdam.

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

[4] J.C.M. Baeten and J.A. Bergstra. Real time process algebra with infinitesimals. In A. Ponse, C. Verhoef, and S.F.M. van Vlijmen, editors, *Algebra of Communicating Processes 1994*, pages 148–187. Workshop in Computing Series, Springer-Verlag, 1995.

[5] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. *Formal Aspects of Computing*, 8(2):188–208, 1996.

[6] J.C.M. Baeten and C.A. Middelburg. Process algebra with timing: Real time and discrete time. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, pages 627–684. Elsevier, 2001.

[7] J.C.M. Baeten and C.A. Middelburg. *Process Algebra with Timing*. To appear in EATCS Monographs Series, Springer-Verlag, 2002.

[8] J.C.M. Baeten and C. Verhoef. Concrete process algebra. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume IV*, pages 149–268. Oxford University Press, 1995.

[9] J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.

[10] J.A. Bergstra and J.W. Klop. The algebra of recursively defined processes and the algebra of regular processes. In J. Paredaens, editor, *Proceedings 11th ICALP*, pages 82–95. LNCS 172, Springer Verlag, 1984.

[11] J.A. Bergstra, C.A. Middelburg, and Y.S. Usenko. Discrete time process algebra and the semantics of SDL. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, pages 1209–1268. Elsevier, 2001.

[12] L. Chen. An interleaving model for real-time systems. In A. Nerode and M. Taitslin, editors, *Symposium on Logical Foundations of Computer Science*, pages 81–92. LNCS 620, Springer-Verlag, 1992.

[13] J. Davies et al. Timed CSP: Theory and practice. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, *Real Time: Theory and Practice*, pages 640–675. LNCS 600, Springer-Verlag, 1992.

[14] M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 117:221–239, 1995.

[15] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR'90*, pages 401–415. LNCS 458, Springer-Verlag, 1990.

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

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

[18] Wang Yi. Real-time behaviour of asynchronous agents. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR'90*, pages 502–520. LNCS 458, Springer-Verlag, 1990.