Fundamenta Informaticae 00 (0000) 1–23
IOS Press

Process Algebra with Nonstandard Timing

C.A. Middelburg

Computing Science Department
Eindhoven University of Technology
Eindhoven, the Netherlands
keesm@win.tue.nl

Department of Philosophy
Utrecht University
Utrecht, the Netherlands
keesm@phil.uu.nl

Abstract. The possibility of two or more actions to be performed consecutively at the same point in time is not excluded in the process algebras from the framework of process algebras with timing presented by Baeten and Middelburg [Handbook of Process Algebra, Elsevier, 2001, Chapter 10]. This possibility is useful in practice when describing and analyzing systems in which actions occur that are entirely independent. However, it is an abstraction of reality to assume that actions can be performed consecutively at the same point in time. In this paper, we propose a process algebra with timing in which this possibility is excluded, but nonstandard non-negative real numbers are included in the time domain. It is shown that this new process algebra generalizes the process algebras with timing from the aforementioned framework in a smooth and natural way.

Keywords: process algebra, timing, nonstandard timing, urgent actions

1. Introduction

In [6], several versions of process algebra with timing, each dealing with timing in a different way and together making up a coherent collection, are introduced. The timing of actions is either relative or

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

absolute and the time scale on which the time is measured is either discrete or continuous. The presented versions originate from the versions of ACP with timing introduced in [1, 2, 3, 4, 5]. However, unlike in the versions with timing presented in [1, 3], execution of actions and passage of time are separated in all versions presented in [6]. Versions in which execution of actions and passage of time are combined in time-stamped actions are regarded as restricted versions. Like in [1, 2, 3], the non-negative real numbers are taken as time domain in the versions with timing on a continuous time scale presented in [6].

Besides, the versions of process algebra with timing on a continuous time scale presented in [6], different from those presented in [1, 3], do not exclude the possibility of two or more actions to be performed consecutively at the same point in time. That is, they include urgent actions, similar to ATP [16] and the different versions of CCS with timing [10, 15, 18]. This feature seems to be essential to obtain simple and natural embeddings of discrete time versions as well as useful in practice when describing and analyzing systems in which actions occur that are entirely independent. This is, for example, the case for actions that happen at different locations in a distributed system. Various systems of that kind are described and analyzed in [7].

In [4], a variant of ACP$\rho\surd$, a version of ACP with timing on a continuous time scale from [2], is introduced that, like the versions presented in [6], does not exclude the possibility of two or more actions to be performed consecutively at the same point in time. In addition, it is shown in [4] that this variant of ACP$\rho\surd$ can be embedded in ACP$\rho\surd$I, which excludes this possibility, by including nonstandard non-negative real numbers in the time domain. ACP$\rho\surd$I is essentially the version of ACP with absolute timing on a continuous time scale from [1] extended with a mechanism for parametric timing, called initial abstraction.

In this paper, we introduce a variant of the most general version of process algebra with timing from [6] that, like the versions from [1, 3], excludes the possibility of two or more actions to be performed consecutively at the same point in time. In this variant, called $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$, nonstandard non-negative real numbers are included in the time domain as well. We further show that the most general version of process algebra with timing from [6], called $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$, can be embedded in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. The given embedding makes clear that $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ generalizes $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ in a smooth and natural way. Hence, the versions of ACP with timing presented in [6] together with $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ still make up a coherent collection. Moreover, the embedding provides a justification of the choice not to exclude in $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ the possibility of two or more actions to be performed consecutively at the same point in time. There is still a lack of experience with applications of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. Consequently, there is no assessment of its usefulness for describing and analyzing systems in this paper.

The main differences between $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ and ACP$\rho\surd$I as presented in [4], with nonstandard non-negative real numbers included in the time domain, are:

- in the former, execution of actions and passage of time are separated, but they are combined in the latter;

- in the former, the time-out operator forces a process to perform its first action before or at the time-out time, but its counterpart in the latter forces a process to perform its first action before the time-out time.

These differences are also main differences between $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ and the embedded variant of ACP$\rho\surd$ from [4]. Moreover, the integration operator is absent from the variant of ACP$\rho\surd$.

$\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ is a version of ACP with absolute timing extended with initial abstraction, like $\mathrm{ACP}\rho\surd\mathrm{I}$. We focus on absolute timing for two reasons. As expounded in [6], relative timing can be expressed by using initial abstraction. And for the purpose of this paper, relative timing has some shortcomings, which are discussed in [14].

The structure of this paper is as follows. In Section 2, $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ is introduced. After that, in Section 3, first $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ is briefly reviewed and then it is shown that $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ generalizes $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ by giving an embedding of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. Finally, in Section 4, some concluding remarks are made. For reference, the axioms of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are given in Appendix A.

2. Process Algebra with Nonstandard Timing

In this section, we introduce a variant of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ that excludes the possibility of two or more actions to be performed consecutively at the same point in time. This variant takes the non-negative finite numbers from a nonstandard model of the real numbers as time domain.

First, we give a brief introduction to nonstandard models of the real numbers. Next, we consider a basic process algebra, called $\mathrm{BPA}^{\mathrm{nsat}}$, that does not cover parallelism and communication. After that, we consider the addition of parallel composition and encapsulation, which results in $\mathrm{ACP}^{\mathrm{nsat}}$. Then, we consider the addition of the integration operator, which results in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$. Integration provides for alternative composition over a continuum of alternatives. Finally, we consider the addition of the initial abstraction operator, which results in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. Initial abstraction is the primary way of forming processes with parametric timing, i.e. processes of which the behaviour depends on their initialization time.

The differences between corresponding axioms of $\mathrm{BPA}^{\mathrm{nsat}}$ and $\mathrm{BPA}^{\mathrm{sat}}$, $\mathrm{ACP}^{\mathrm{nsat}}$ and $\mathrm{ACP}^{\mathrm{sat}}$, etc. are informally discussed at the appropriate places, and so are the differences with corresponding axioms of BPA_δ and ACP. BPA_δ is the basic process algebra in the case of ACP. BPA is the subtheory of BPA_δ without the constant δ. For more information on ACP (without timing), see [8, 9].

For $\mathrm{BPA}^{\mathrm{nsat}}$, $\mathrm{ACP}^{\mathrm{nsat}}$ and $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$, a structural operational semantics is given by means of transition rules. For more information on the approach followed in this paper, including the use of negative premises in transition rules and the use of second-order variables in the presence of variable binding operators, see [12, 13].

2.1. Nonstandard Models of the Real Numbers

A *nonstandard* model of the real numbers has the following general properties:

- there is a set $\mathbb{R}^*$ such that $\mathbb{R} \subset \mathbb{R}^*$;
- for each n-ary function f on $\mathbb{R}$, including addition, multiplication, additive inverse and multiplicative inverse, there is a corresponding n-ary function f^* on $\mathbb{R}^*$ that agrees on $\mathbb{R}$;
- for each n-ary relation P on $\mathbb{R}$, including ordering, there is a corresponding n-ary relation P^* on $\mathbb{R}^*$ that agrees on $\mathbb{R}$;
- there is an element of $\mathbb{R}^*$ that is greater than 0 and smaller than all positive elements of $\mathbb{R}$ according to the ordering relation on $\mathbb{R}^*$;

- each statement ϕ in which the range of quantified variables is made explicit is true for $\mathbb{R}$ if and only if the statement ϕ^*, obtained from ϕ by replacing each f by f^* and each P by P^*, is true for $\mathbb{R}^*$.

An $s \in \mathbb{R}^*$ is a non-negative *finite* number if $0 \leq s \leq p$ for some non-negative $p \in \mathbb{R}$. An $s \in \mathbb{R}^*$ is a positive *infinitesimal* number if $0 < s < p$ for all positive $p \in \mathbb{R}$. The non-negative finite numbers contain the standard non-negative real numbers with nonstandard non-negative real numbers clustered infinitely close to each standard non-negative real number.

We use the following notations. We write $\mathbb{R}^{\geq}$ for the set of non-negative real numbers. We use $p, p', \ldots$ to stand for arbitrary elements of $\mathbb{R}^{\geq}$, and $V, W, \ldots$ to stand for arbitrary subsets of $\mathbb{R}^{\geq}$. We write $\mathbb{F}$ for the set of non-negative finite numbers, and $\mathbb{I}$ for the set of positive infinitesimal numbers. We use $s, s', \ldots$ to stand for arbitrary elements of $\mathbb{F}$, $\epsilon, \epsilon', \ldots$ to stand for arbitrary elements of $\mathbb{I}$, and $S, T, \ldots$ to stand for arbitrary subsets of $\mathbb{F}$.

We do not fix a particular nonstandard model of the real numbers. All of them have the necessary properties. In addition to the properties inherited from the ordinary real numbers, the following properties are basic ones:

- $\mathbb{R}^{\geq} \subset \mathbb{F}$;
- for all $s, s' \in \mathbb{F}$, $s + s' \in \mathbb{F}$ and $s \cdot s' \in \mathbb{F}$;
- for all $\epsilon, \epsilon' \in \mathbb{I}$, $\epsilon + \epsilon' \in \mathbb{I}$ and $\epsilon \cdot \epsilon' \in \mathbb{I}$;
- for all $s \in \mathbb{F} \setminus (\mathbb{I} \cup \{0\})$, $s^{-1} \in \mathbb{F} \setminus (\mathbb{I} \cup \{0\})$;
- for all $s \in \mathbb{F} \setminus (\mathbb{I} \cup \{0\})$ and $\epsilon \in \mathbb{I}$, $s + \epsilon \in \mathbb{F} \setminus (\mathbb{I} \cup \{0\})$ and $s \cdot \epsilon \in \mathbb{I}$;
- for all $s \in \mathbb{F}$, there exists a unique $p \in \mathbb{R}^{\geq}$ such that there exists an $\epsilon \in \mathbb{I} \cup \{0\}$ such that $p + \epsilon = s$.

For $s, s' \in \mathbb{F}$, we write $s \approx s'$ to indicate that there is an $\epsilon \in \mathbb{I} \cup \{0\}$ such that $s + \epsilon = s'$ or $s = s' + \epsilon$. In the case where $s \approx s'$, we say that s is *infinitely close* to s'. For $s \in \mathbb{F}$, we write s° for the unique $p \in \mathbb{R}^{\geq}$ such that $p \approx s$. We call s° the *standard part* of s. For $S \subseteq \mathbb{F}$, we write S° for the set $\{v^\circ \mid v \in S\}$. The most frequently used properties concerning standard parts are:

- for all $s, s' \in \mathbb{F}$, if $s \leq s'$ then $s^\circ \leq s'^\circ$;
- for all $s, s' \in \mathbb{F}$, $s \approx s'$ if and only if $s^\circ = s'^\circ$;
- for all $s, s' \in \mathbb{F}$, $(s + s')^\circ = s^\circ + s'^\circ$ and $(s \cdot s')^\circ = s^\circ \cdot s'^\circ$;
- for all $s \in \mathbb{F}$, $s^\circ \leq s$; for all $\epsilon \in \mathbb{I}$, $\epsilon^\circ = 0$; for all $p \in \mathbb{R}^{\geq}$, $p^\circ = p$.

For more information on nonstandard models of the real numbers, see [11, 17].

2.2. Basic Process Algebra

The atomic processes are undelayable actions. Let a be an action. Then *undelayable action* a, written $\hat{a}$, is the process that performs action a at point of time 0 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 absolute delay. Let p be a process and $u \in \mathbb{F}$. Then the *absolute delay* of p for a period of u time units, written $\sigma^u_{\mathsf{abs}}(p)$, is the process that idles a period of u time units longer than p and otherwise behaves like p. That is, p is delayed a period of u time units. For example, $\sigma^3_{\mathsf{abs}}(\hat{a})$ is the process that idles from point of time 0 till point of time 3, and at that point of time first performs action a and then terminates successfully – like the time-stamped action $a(3)$ from [1].

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. For example, $\sigma^3_{\mathsf{abs}}(\hat{a}) + \sigma^1_{\mathsf{abs}}(\hat{b})$ is a process that idles from point of time 0 till point of time 1, and at that point of time either (i) first performs action b and then terminates successfully or (ii) idles further till point of time 3, and at that point of time first performs action a and then terminates successfully. 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. Notice that, from the point of successful termination of p_1, only the part of p_2 that idles till after that point still counts. If p_1 terminates successfully and no part of p_2 can idle till after the point of successful termination, the sequential composition of p_1 and p_2 will at that point become inactive – not being capable of performing any action nor being capable of idling. For example, both $\sigma^3_{\mathsf{abs}}(\hat{a}) \cdot \sigma^1_{\mathsf{abs}}(\hat{b})$ and $\sigma^3_{\mathsf{abs}}(\hat{a}) \cdot \sigma^3_{\mathsf{abs}}(\hat{b})$ are processes that idle from point of time 0 till point of time 3, and at that point of time first perform action a and then become inactive.

In order to deal with processes that become inactive, we need an additional process that is neither capable of performing any action nor capable of idling beyond point of time 0. This process, written $\hat{\delta}$, is called *undelayable deadlock*. Hence, $\sigma^3_{\mathsf{abs}}(\hat{a}) \cdot \sigma^1_{\mathsf{abs}}(\hat{a})$ and $\sigma^3_{\mathsf{abs}}(\hat{a}) \cdot \sigma^3_{\mathsf{abs}}(\hat{a})$ behave the same as $\sigma^3_{\mathsf{abs}}(\hat{a}) \cdot \sigma^3_{\mathsf{abs}}(\hat{\delta})$. This example shows that, unlike in the versions of ACP with timing presented in [6], there is no need for a process that can be viewed as a trace of a process that has become inactive before point of time 0 (represented by $\dot{\delta}$ in [6]) to handle situations in which processes exhibit inconsistent timing. This is due to the exclusion of the possibility of two or more actions to be performed consecutively at the same point in time: if a process becomes inactive at the point of time that it performs its last action, this indicates that it becomes inactive for a timing inconsistency. Besides by timing inconsistencies, a process can become inactive by incapability to communicate or encapsulation, as explained in Section 2.3. However, in those other cases, the process will never become inactive at the point of time that it performs its last action.

In order to capture timing fully, we have, in addition to absolute delay, nonstandard absolute time-out and nonstandard absolute initialization. Let p be a process and $u \in \mathbb{F}$. The *nonstandard absolute time-out* of p at point of time u, written $\hat{\upsilon}^u_{\mathsf{abs}}(p)$, behaves either like the part of p that does not idle till after point of time u or like undelayable deadlock delayed a period of time u if p is capable of idling till after point of time u. Otherwise, it behaves like p. That is, the nonstandard absolute time-out keeps p from idling till after point of time u. The *nonstandard absolute initialization* of p at point of time u, written $\hat{\overline{\upsilon}}^u_{\mathsf{abs}}(p)$, behaves like the part of p that idles till after point of time u if p is capable of idling till after point of time u. Otherwise, it behaves like undelayable deadlock delayed a period of time u. That is, the nonstandard absolute initialization keeps p from performing actions before or at point of time u. Thus, if p is capable of idling till after point of time u, p behaves the same as $\hat{\upsilon}^u_{\mathsf{abs}}(p) + \hat{\overline{\upsilon}}^u_{\mathsf{abs}}(p)$. For example, $\hat{\upsilon}^2_{\mathsf{abs}}(\sigma^3_{\mathsf{abs}}(\hat{a}) + \sigma^1_{\mathsf{abs}}(\hat{b}))$ behaves the same as $\sigma^1_{\mathsf{abs}}(\hat{b}) + \sigma^2_{\mathsf{abs}}(\hat{\delta})$ and $\hat{\overline{\upsilon}}^2_{\mathsf{abs}}(\sigma^3_{\mathsf{abs}}(\hat{a}) + \sigma^1_{\mathsf{abs}}(\hat{b}))$ behaves the same as $\sigma^3_{\mathsf{abs}}(\hat{a})$.

Although the possibility of two or more actions to be performed consecutively at the same point in time is excluded, we can have that they are performed at points in time that are infinitely close to one another. In the extended time domain $\mathbb{F}$, actions that are performed consecutively at the same point of time in the time domain $\mathbb{R}^{\geq}$, say p, can be considered to be performed at different points of time that are infinitely close to p. Thus, the exclusion of the possibility of two or more actions to be performed consecutively at the same point in time is fully compensated by taking the non-negative finite numbers from a nonstandard model of the real numbers as time domain.

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

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. Throughout this paper 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, (iii) all other operators have the same precedence.

Axioms of BPA$^{\mathrm{nsat}}$

The axiom system of BPA$^{\mathrm{nsat}}$ consists of the equations given in Table 1. Axioms A1–A5 are the axioms of BPA. Axioms A6NSA and A7NSA are simple reformulations of axioms A6 and A7 of BPA$_\delta$. Axioms NSAT1 and NSAT2 point out that a delay of 0 time units has no effect and that consecutive delays count up. Axiom NSAT3, called the time-factorization axiom, shows that a delay by itself cannot determine a choice. Axioms NSAT4 and NSAT5 reflect that timing is absolute and that the possibility of two or more actions to be performed consecutively at the same point in time is excluded. These axioms become easier to understand by realizing the following.

Lemma 2.1. For all closed BPA$^{\mathrm{nsat}}$-terms t and for all $s \in \mathbb{F}$ either there exists a closed BPA$^{\mathrm{nsat}}$-term t' such that $t = \hat{\upsilon}^s_{\mathsf{abs}}(t')$ is derivable or there exist closed BPA$^{\mathrm{nsat}}$-terms t' and t'' such that $t = \hat{\upsilon}^s_{\mathsf{abs}}(t') + \sigma^s_{\mathsf{abs}}(t'')$ is derivable.

Proof:
Straightforward by induction on the structure of t. □

Axioms NSATO0–NSATO5 and NSAI0–NSAI5 are the defining equations of the nonstandard absolute time-out operator and nonstandard absolute initialization operator, respectively. These axioms reflect the intended meaning of these operators clearly. Axioms NSATO1, NSATO2, NSAI1 and NSAI2 are quite different from axioms SATO1, SATO2, SAI1 and SAI2 of BPA$^{\mathrm{sat}}$ because the time-out and initialization

Table 1. Axioms of $\mathrm{BPA}^{\mathrm{nsat}}$ ($a \in \mathsf{A}_\delta,\ s,t \geq 0,\ u > 0$)

$x + y = y + x$	A1	$\sigma^{0}_{\mathsf{abs}}(\hat{a}) = \hat{a}$	NSAT1
$(x + y) + z = x + (y + z)$	A2	$\sigma^{s}_{\mathsf{abs}}(\sigma^{t}_{\mathsf{abs}}(x)) = \sigma^{s+t}_{\mathsf{abs}}(x)$	NSAT2
$x + x = x$	A3	$\sigma^{s}_{\mathsf{abs}}(x) + \sigma^{s}_{\mathsf{abs}}(y) = \sigma^{s}_{\mathsf{abs}}(x + y)$	NSAT3
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4	$\sigma^{s}_{\mathsf{abs}}(x) \cdot \hat{\upsilon}^{s}_{\mathsf{abs}}(y) = \sigma^{s}_{\mathsf{abs}}(x \cdot \hat{\delta})$	NSAT4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5	$\sigma^{s}_{\mathsf{abs}}(x) \cdot (\hat{\upsilon}^{s}_{\mathsf{abs}}(y) + \sigma^{s}_{\mathsf{abs}}(z)) =$	
$x + \hat{\delta} = x$	A6NSA	$\sigma^{s}_{\mathsf{abs}}(x \cdot \sigma^{0}_{\mathsf{abs}}(z))$	NSAT5
$\hat{\delta} \cdot x = \hat{\delta}$	A7NSA		
$\hat{\upsilon}^{0}_{\mathsf{abs}}(\sigma^{u}_{\mathsf{abs}}(x)) = \hat{\delta}$	NSATO1	$\hat{\overline{\upsilon}}^{0}_{\mathsf{abs}}(\sigma^{u}_{\mathsf{abs}}(x)) = \sigma^{u}_{\mathsf{abs}}(x)$	NSAI1
$\hat{\upsilon}^{s}_{\mathsf{abs}}(\hat{a}) = \hat{a}$	NSATO2	$\hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(\hat{a}) = \sigma^{s}_{\mathsf{abs}}(\hat{\delta})$	NSAI2
$\hat{\upsilon}^{s+t}_{\mathsf{abs}}(\sigma^{s}_{\mathsf{abs}}(x)) = \sigma^{s}_{\mathsf{abs}}(\hat{\upsilon}^{t}_{\mathsf{abs}}(x))$	NSATO3	$\hat{\overline{\upsilon}}^{s+t}_{\mathsf{abs}}(\sigma^{s}_{\mathsf{abs}}(x)) = \sigma^{s}_{\mathsf{abs}}(\hat{\overline{\upsilon}}^{t}_{\mathsf{abs}}(\sigma^{0}_{\mathsf{abs}}(x)))$	NSAI3
$\hat{\upsilon}^{s}_{\mathsf{abs}}(x + y) = \hat{\upsilon}^{s}_{\mathsf{abs}}(x) + \hat{\upsilon}^{s}_{\mathsf{abs}}(y)$	NSATO4	$\hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(x + y) = \hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(x) + \hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(y)$	NSAI4
$\hat{\upsilon}^{s}_{\mathsf{abs}}(x \cdot y) = \hat{\upsilon}^{s}_{\mathsf{abs}}(x) \cdot y$	NSATO5	$\hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(x \cdot y) = \hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(x) \cdot y$	NSAI5

operators differ in detail. In the case of $\mathrm{BPA}^{\mathrm{sat}}$, the time-out operator forces a process to perform its first action before the time-out time and, consistently, the initialization operator forces a process not to perform its first action before the initialization time. Actually, axioms NSAT4 and NSAT5 remain, despite the exclusion of the possibility of two or more actions to be performed consecutively at the same point in time, simple reformulations of axioms SAT4 and SAT5 because of the different time-out operators. The following natural counterparts of axioms NSATO3 and NSAI3 are easily derivable from the axioms of $\mathrm{BPA}^{\mathrm{nsat}}$:

$$\hat{\upsilon}^{s}_{\mathsf{abs}}(\sigma^{s+u}_{\mathsf{abs}}(x)) = \sigma^{s}_{\mathsf{abs}}(\hat{\delta}) \qquad \mathrm{NSATO3}'$$
$$\hat{\overline{\upsilon}}^{s}_{\mathsf{abs}}(\sigma^{s+u}_{\mathsf{abs}}(x)) = \sigma^{s+u}_{\mathsf{abs}}(x) \qquad \mathrm{NSAI3}'$$

Except for axioms NSAT1, NSAT5, NSATO1, NSATO2, NSAI1, NSAI2 and NSAI3, the axioms of $\mathrm{BPA}^{\mathrm{nsat}}$ are simple reformulations of axioms of $\mathrm{BPA}^{\mathrm{sat}}$. However, we could have taken axioms similar to NSAT1, NSAT5 and NSAI3 for SAT1, SAT5 and SAI3: for closed terms, the derivable equations would remain the same. We already explained above why axioms NSATO1, NSATO2, NSAI1 and NSAI2 are no simple reformulations. The absence of a counterpart of the constant $\dot{\delta}$ in $\mathrm{BPA}^{\mathrm{nsat}}$ makes the whole axiom system less extensive than the axiom system of $\mathrm{BPA}^{\mathrm{sat}}$.

Semantics of $\mathrm{BPA}^{\mathrm{nsat}}$

The structural operational semantics of $\mathrm{BPA}^{\mathrm{nsat}}$ is described by the rules given in Table 2. The following transition predicates are used in Table 2: a binary predicate $\langle\_\,, s\rangle \xrightarrow{a} \langle\_\,, s\rangle$ and a unary predicate $\langle\_\,, s\rangle \xrightarrow{a} \langle\surd, s\rangle$ for each $a \in \mathsf{A}$ and $s \in \mathbb{F}$, and a binary predicate $\langle\_\,, s\rangle \overset{u}{\longmapsto} \langle\_\,, s'\rangle$ for each $s, s', u \in \mathbb{F}$ such that $s + u = s'$ and $u > 0$. The transition predicates can be explained as follows:

Table 2. Rules for operational semantics of $\mathrm{BPA}^{\mathrm{nsat}}$ ($a \in \mathsf{A}$, $s, t \geq 0$, $u > 0$)

$$\frac{}{\langle \hat{a}, 0\rangle \xrightarrow{a} \langle \surd, 0\rangle}$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \sigma^{0}_{\mathsf{abs}}(x), s\rangle \xrightarrow{a} \langle x', s\rangle} \quad \frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \sigma^{u}_{\mathsf{abs}}(x), s+u\rangle \xrightarrow{a} \langle \sigma^{u}_{\mathsf{abs}}(x'), s+u\rangle} \quad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle \sigma^{t}_{\mathsf{abs}}(x), s+t\rangle \xrightarrow{a} \langle \surd, s+t\rangle}$$

$$\frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle}{\langle \sigma^{t}_{\mathsf{abs}}(x), s+t\rangle \overset{u}{\longmapsto} \langle \sigma^{t}_{\mathsf{abs}}(x), s+t+u\rangle} \quad \frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle}{\langle \sigma^{t}_{\mathsf{abs}}(x), s\rangle \overset{t+u}{\longmapsto} \langle \sigma^{t}_{\mathsf{abs}}(x), s+t+u\rangle}$$

$$\frac{}{\langle \sigma^{t+u}_{\mathsf{abs}}(x), s\rangle \overset{u}{\longmapsto} \langle \sigma^{t+u}_{\mathsf{abs}}(x), s+u\rangle}\; t \geq s$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle x+y, s\rangle \xrightarrow{a} \langle x', s\rangle} \quad \frac{\langle y, s\rangle \xrightarrow{a} \langle y', s\rangle}{\langle x+y, s\rangle \xrightarrow{a} \langle y', s\rangle} \quad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle x+y, s\rangle \xrightarrow{a} \langle \surd, s\rangle} \quad \frac{\langle y, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle x+y, s\rangle \xrightarrow{a} \langle \surd, s\rangle}$$

$$\frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle}{\langle x+y, s\rangle \overset{u}{\longmapsto} \langle x+y, s+u\rangle} \quad \frac{\langle y, s\rangle \overset{u}{\longmapsto} \langle y, s+u\rangle}{\langle x+y, s\rangle \overset{u}{\longmapsto} \langle x+y, s+u\rangle}$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle x \cdot y, s\rangle \xrightarrow{a} \langle x' \cdot y, s\rangle} \quad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle x \cdot y, s\rangle \xrightarrow{a} \langle \hat{\bar{\upsilon}}^{s}_{\mathsf{abs}}(y), s\rangle} \quad \frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle}{\langle x \cdot y, s\rangle \overset{u}{\longmapsto} \langle x \cdot y, s+u\rangle}$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \hat{\upsilon}^{t}_{\mathsf{abs}}(x), s\rangle \xrightarrow{a} \langle x', s\rangle}\; t \geq s \quad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle \hat{\upsilon}^{t}_{\mathsf{abs}}(x), s\rangle \xrightarrow{a} \langle \surd, s\rangle}\; t \geq s \quad \frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle}{\langle \hat{\upsilon}^{t}_{\mathsf{abs}}(x), s\rangle \overset{u}{\longmapsto} \langle \hat{\upsilon}^{t}_{\mathsf{abs}}(x), s+u\rangle}\; t \geq s+u$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \hat{\bar{\upsilon}}^{t}_{\mathsf{abs}}(x), s\rangle \xrightarrow{a} \langle x', s\rangle}\; t < s \quad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle \hat{\bar{\upsilon}}^{t}_{\mathsf{abs}}(x), s\rangle \xrightarrow{a} \langle \surd, s\rangle}\; t < s \quad \frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle}{\langle \hat{\bar{\upsilon}}^{t}_{\mathsf{abs}}(x), s\rangle \overset{u}{\longmapsto} \langle \hat{\bar{\upsilon}}^{t}_{\mathsf{abs}}(x), s+u\rangle}\; t < s+u$$

$$\frac{}{\langle \hat{\bar{\upsilon}}^{t+u}_{\mathsf{abs}}(x), s\rangle \overset{u}{\longmapsto} \langle \hat{\bar{\upsilon}}^{t+u}_{\mathsf{abs}}(x), s+u\rangle}\; t \geq s$$

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

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

$\langle t, s\rangle \overset{u}{\longmapsto} \langle t, s'\rangle$: process t is capable of first idling from point of time s to point of time s' and then proceeding as process t.

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

Notice the second transition rule for sequential composition. The nonstandard absolute initialization operator in the conclusion enforces that no more actions are performed at time s. This would not be the case with an absolute initialization operator that is essentially the same as the one of $\mathrm{BPA}^{\mathrm{sat}}$.

Bisimulation based on the transition rules for $\mathrm{BPA}^{\mathrm{nsat}}$ is defined as usual. The quotient algebra of the algebra of closed terms over the signature of $\mathrm{BPA}^{\mathrm{nsat}}$ by bisimulation equivalence is a model of the axioms of $\mathrm{BPA}^{\mathrm{nsat}}$. For more information on the construction of such bisimulation models, see e.g. [7].

2.3. Algebra of Communicating Processes

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: (i) 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; (ii) 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 (i) performing an action of p_1 or p_2 if it can do so before the ultimate point of time for the other process to start performing actions or to become inactive; (ii) 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. 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 handshaking 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. The process $(\sigma^2_{\mathsf{abs}}(\widehat{s_1(0)}) + \sigma^3_{\mathsf{abs}}(\widehat{s_1(0)})) \parallel \sigma^3_{\mathsf{abs}}(\widehat{r_1(0)})$ behaves the same as $\sigma^2_{\mathsf{abs}}(\widehat{s_1(0)}) \cdot \sigma^3_{\mathsf{abs}}(\widehat{r_1(0)}) + \sigma^3_{\mathsf{abs}}(\widehat{c_1(0)})$. This example shows that 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 inactive 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. For example, the process $\partial_{\{s_1(0), r_1(0)\}}((\sigma^2_{\mathsf{abs}}(\widehat{s_1(0)}) + \sigma^3_{\mathsf{abs}}(\widehat{s_1(0)})) \parallel \sigma^3_{\mathsf{abs}}(\widehat{r_1(0)}))$ behaves the same as $\sigma^3_{\mathsf{abs}}(\widehat{c_1(0)})$.

In order to obtain a finite axiomatization of parallel composition, we also have the usual auxiliary operators for that purpose: $\mathbin{\|\!\!\_}$ and $|$. The operator $\mathbin{\|\!\!\_}$ is interpreted as left merge, which is the same as parallel composition except that the left merge of p_1 and p_2 starts 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 starts with performing an action of p_1 and an action of p_2 synchronously.

It is assumed that a fixed but arbitrary *communication function*, i.e. a partial commutative and associative 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.

Axioms of ACP$^{\mathrm{nsat}}$

The axiom system of ACP$^{\mathrm{nsat}}$ consists of the axioms of BPA$^{\mathrm{nsat}}$ and the equations given in Table 3. Axioms CM1, CM4, CM8, CM9, D3 and D4 are in common with ACP. Axioms CM2NSA, CM3NSA, CM5NSA–CM7NSA, CF1NSA, CF2NSA, D1NSA and D2NSA are simple reformulations of axioms CM2, CM3, CM5–CM7, CF1, CF2, D1 and D2 of ACP. Axioms NSACM1–NSACM5 and NSAD are new axioms concerning the interaction of absolute delay with left merge, communication merge and

Table 3. Additional axioms for ACP$^{\mathrm{nsat}}$ $(a, b \in \mathsf{A}_\delta,\ c \in \mathsf{A},\ s \geq 0,\ u > 0)$

$x \parallel y = x \mathbin{⌊\!⌊} y + y \mathbin{⌊\!⌊} x + x \mid y$	CM1	$\hat{a} \cdot x \mid \hat{b} = (\hat{a} \mid \hat{b}) \cdot x$	CM5NSA
$\hat{a} \mathbin{⌊\!⌊} x = \hat{a} \cdot x$	CM2NSA	$\hat{a} \mid \hat{b} \cdot x = (\hat{a} \mid \hat{b}) \cdot x$	CM6NSA
$\hat{a} \cdot x \mathbin{⌊\!⌊} y = \hat{a} \cdot (x \parallel y)$	CM3NSA	$\hat{a} \cdot x \mid \hat{b} \cdot y = (\hat{a} \mid \hat{b}) \cdot (x \parallel y)$	CM7NSA
$\sigma^{s}_{\mathsf{abs}}(x) \mathbin{⌊\!⌊} \hat{\upsilon}^{0}_{\mathsf{abs}}(y) = \hat{\delta}$	NSACM1	$\hat{\upsilon}^{0}_{\mathsf{abs}}(x) \mid \sigma^{u}_{\mathsf{abs}}(y) = \hat{\delta}$	NSACM3
$\sigma^{s}_{\mathsf{abs}}(x) \mathbin{⌊\!⌊} (\hat{\upsilon}^{s}_{\mathsf{abs}}(y) + \sigma^{s+u}_{\mathsf{abs}}(z)) =$		$\sigma^{u}_{\mathsf{abs}}(x) \mid \hat{\upsilon}^{0}_{\mathsf{abs}}(y) = \hat{\delta}$	NSACM4
$\sigma^{s}_{\mathsf{abs}}(x \mathbin{⌊\!⌊} \sigma^{u}_{\mathsf{abs}}(z))$	NSACM2	$\sigma^{s}_{\mathsf{abs}}(x) \mid \sigma^{s}_{\mathsf{abs}}(y) = \sigma^{s}_{\mathsf{abs}}(x \mid y)$	NSACM5
$(x + y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} z + y \mathbin{⌊\!⌊} z$	CM4	$(x + y) \mid z = x \mid z + y \mid z$	CM8
		$x \mid (y + z) = x \mid y + x \mid z$	CM9
$\partial_H(\hat{a}) = \hat{a}$ if $a \notin H$	D1NSA	$\hat{a} \mid \hat{b} = \hat{c}$ if $\gamma(a, b) = c$	CF1NSA
$\partial_H(\hat{a}) = \hat{\delta}$ if $a \in H$	D2NSA	$\hat{a} \mid \hat{b} = \hat{\delta}$ if $\gamma(a, b)$ undefined	CF2NSA
$\partial_H(\sigma^{s}_{\mathsf{abs}}(x)) = \sigma^{s}_{\mathsf{abs}}(\partial_H(x))$	NSAD		
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D3		
$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$	D4		

encapsulation. Axioms NSACM1–NSACM5 become easier to understand by realizing the following.

Lemma 2.2. For all closed ACP$^{\mathrm{nsat}}$-terms t either there exists a closed ACP$^{\mathrm{nsat}}$-term t' such that $t = \hat{\upsilon}^{0}_{\mathsf{abs}}(t')$ is derivable or there exists a $s' \in \mathbb{F}$ such that for all $s \in \mathbb{F}$ with $s \leq s'$ there exist a $u \in \mathbb{F}$ with $u > 0$ and closed ACP$^{\mathrm{nsat}}$-terms t' and t'' such that $t = \hat{\upsilon}^{s}_{\mathsf{abs}}(t') + \sigma^{s+u}_{\mathsf{abs}}(t'')$ is derivable.

Proof:
Straightforward by induction on the structure of t. □

Except for axiom NSACM2, the additional axioms for ACP$^{\mathrm{nsat}}$ are simple reformulations of axioms of ACP$^{\mathrm{sat}}$. However, unlike in axioms SACM1, SACM3 and SACM4, no new auxiliary operator is needed in axioms NSACM1, NSACM3 and NSACM4. This is due to the different time-out operators. Besides, axioms NSACM1, NSACM3 and NSACM4 are simpler than axioms SACM1, SACM3 and SACM4 because of the absence of a counterpart of the constant $\dot{\delta}$ in ACP$^{\mathrm{nsat}}$. Actually, the absence of this constant makes the whole axiom system less extensive than the axiom system of ACP$^{\mathrm{sat}}$. Axiom NSACM2 is slightly different from axiom SACM2 of ACP$^{\mathrm{sat}}$ because of the exclusion of the possibility of two or more actions to be performed consecutively at the same point in time.

Semantics of ACP$^{\mathrm{nsat}}$

The structural operational semantics of ACP$^{\mathrm{nsat}}$ is described by the rules for BPA$^{\mathrm{nsat}}$ and the rules given in Table 4.

Notice that like before in the conclusion of the first four transition rules for parallel composition and the first two transition rules for left merge the nonstandard absolute initialization operator is used to enforce that no more actions are performed at time s.

A bisimulation model of the axioms of ACP$^{\mathrm{nsat}}$ can be constructed in the same way as for BPA$^{\mathrm{nsat}}$.

Table 4. Additional rules for ACP$^{\mathrm{nsat}}$ ($a, b, c \in \mathsf{A}$, $s \geq 0$, $u > 0$)

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle,\ \langle y, 0\rangle \overset{u}{\longmapsto} \langle y, u\rangle}{\langle x \parallel y, s\rangle \xrightarrow{a} \langle x' \parallel \hat{\bar{v}}^{s}_{\mathsf{abs}}(y), s\rangle}\ u > s \qquad \frac{\langle x, 0\rangle \overset{u}{\longmapsto} \langle x, u\rangle,\ \langle y, s\rangle \xrightarrow{a} \langle y', s\rangle}{\langle x \parallel y, s\rangle \xrightarrow{a} \langle \hat{\bar{v}}^{s}_{\mathsf{abs}}(x) \parallel y', s\rangle}\ u > s$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle,\ \langle y, 0\rangle \overset{u}{\longmapsto} \langle y, u\rangle}{\langle x \parallel y, s\rangle \xrightarrow{a} \langle \hat{\bar{v}}^{s}_{\mathsf{abs}}(y), s\rangle}\ u > s \qquad \frac{\langle x, 0\rangle \overset{u}{\longmapsto} \langle x, u\rangle,\ \langle y, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle x \parallel y, s\rangle \xrightarrow{a} \langle \hat{\bar{v}}^{s}_{\mathsf{abs}}(x), s\rangle}\ u > s$$

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

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

$$\frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s + u\rangle,\ \langle y, s\rangle \overset{u}{\longmapsto} \langle y, s + u\rangle}{\langle x \parallel y, s\rangle \overset{u}{\longmapsto} \langle x \parallel y, s + u\rangle}$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle,\ \langle y, 0\rangle \overset{u}{\longmapsto} \langle y, u\rangle}{\langle x \mathbin{\lfloor\!\lfloor} y, s\rangle \xrightarrow{a} \langle x' \parallel \hat{\bar{v}}^{s}_{\mathsf{abs}}(y), s\rangle}\ u > s \qquad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle,\ \langle y, 0\rangle \overset{u}{\longmapsto} \langle y, u\rangle}{\langle x \mathbin{\lfloor\!\lfloor} y, s\rangle \xrightarrow{a} \langle \hat{\bar{v}}^{s}_{\mathsf{abs}}(y), s\rangle}\ u > s$$

$$\frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s + u\rangle,\ \langle y, s\rangle \overset{u}{\longmapsto} \langle y, s + u\rangle}{\langle x \mathbin{\lfloor\!\lfloor} y, s\rangle \overset{u}{\longmapsto} \langle x \mathbin{\lfloor\!\lfloor} y, s + u\rangle}$$

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

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

$$\frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s + u\rangle,\ \langle y, s\rangle \overset{u}{\longmapsto} \langle y, s + u\rangle}{\langle x \mid y, s\rangle \overset{u}{\longmapsto} \langle x \mid y, s + u\rangle}$$

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \partial_H(x), s\rangle \xrightarrow{a} \langle \partial_H(x'), s\rangle}\ a \notin H \qquad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle \partial_H(x), s\rangle \xrightarrow{a} \langle \surd, s\rangle}\ a \notin H \qquad \frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s + u\rangle}{\langle \partial_H(x), s\rangle \overset{u}{\longmapsto} \langle \partial_H(x), s + u\rangle}$$

2.4. 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 ACP$^{\mathrm{nsat}}$. Integration is represented by the variable-binding operator $\int$. Let P be an expression, possibly containing variable v, such that $P[s/v]$ (P with s substituted for v) represents a process for all $s \in \mathbb{F}$; and let $S \subseteq \mathbb{F}$. Then the *integration* $\int_{v \in S} P$ behaves like one of the processes $P[s/v]$ for $s \in S$. Hence, integration is a form of alternative composition over a set of alternatives that may even be a continuum. For example, $\int_{v \in [0,3)} \sigma^{v}_{\mathsf{abs}}(\hat{a})$ is a process that idles from point of time 0 till an arbitrary point of time less than 3 and at that point of time first performs action a and then terminates successfully.

Axioms of ACP$^{\mathrm{nsat}}$I

The axiom system of ACP$^{\mathrm{nsat}}$I consists of the axioms of ACP$^{\mathrm{nsat}}$ and the equations given in Table 5. We use F and G as variables ranging over functions that map each $s \in \mathbb{F}$ to a process and can be represented by terms containing a designated free variable ranging over $\mathbb{F}$. For more information on such second-order variables, see e.g. [12, 13]. Axiom INT1 is similar to the α-conversion rule of λ-

Table 5. Additional axioms for $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$ ($s \geq 0$, $u > 0$ and $u \notin \mathbb{I}$)

$\int_{v \in S} F(v) = \int_{w \in S} F(w)$	INT1	$\sup S = s, s \in S \Rightarrow \int_{v \in S} \sigma^{v^\circ}_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{\delta})) = \sigma^{s^\circ}_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{\delta}))$	INT9NSA
$\int_{v \in \emptyset} F(v) = \hat{\delta}$	INT2	$\int_{v \in S} \sigma^{s}_{\mathsf{abs}}(F(v)) = \sigma^{s}_{\mathsf{abs}}(\int_{v \in S} F(v))$	INT10NSA
$\int_{v \in \{s\}} F(v) = F(s)$	INT3	$\int_{v \in S}(F(v) + G(v)) = \int_{v \in S} F(v) + \int_{v \in S} G(v)$	INT11
$\int_{v \in S \cup T} F(v) = \int_{v \in S} F(v) + \int_{v \in T} F(v)$	INT4	$\int_{v \in S}(F(v) \cdot x) = (\int_{v \in S} F(v)) \cdot x$	INT12
$S \neq \emptyset \Rightarrow \int_{v \in S} x = x$	INT5	$\int_{v \in S}(F(v) \mathbin{\|\!\lfloor} x) = (\int_{v \in S} F(v)) \mathbin{\|\!\lfloor} x$	INT13
$(\forall v \in S \bullet F(v) = G(v)) \Rightarrow \int_{v \in S} F(v) = \int_{v \in S} G(v)$	INT6	$\int_{v \in S}(F(v) \mid x) = (\int_{v \in S} F(v)) \mid x$	INT14
$\sup S = s \Rightarrow \int_{v \in S} \sigma^{v}_{\mathsf{abs}}(\hat{\delta}) = \sigma^{s}_{\mathsf{abs}}(\hat{\delta})$	INT7NSAa	$\int_{v \in S}(x \mid F(v)) = x \mid (\int_{v \in S} F(v))$	INT15
S, T unbounded $\Rightarrow \int_{v \in S} \sigma^{v}_{\mathsf{abs}}(\hat{\delta}) = \int_{v \in T} \sigma^{v}_{\mathsf{abs}}(\hat{\delta})$	INT7NSAb	$\int_{v \in S} \partial_H(F(v)) = \partial_H(\int_{v \in S} F(v))$	INT16
$\sup S = s, s \notin S \Rightarrow \int_{v \in S} \sigma^{v^\circ}_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{\delta})) = \sigma^{s^\circ}_{\mathsf{abs}}(\hat{\delta})$	INT8NSAa	$\int_{v \in S^\circ} F(v) = \int_{v \in S} F(v^\circ)$	NSAINT
S, T unbounded $\Rightarrow \int_{v \in S} \sigma^{v}_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{\delta})) = \int_{v \in T} \sigma^{v}_{\mathsf{abs}}(\hat{\delta})$	INT8NSAb	$\hat{\upsilon}^{s}_{\mathsf{abs}}(\int_{v \in S} F(v)) = \int_{v \in S} \hat{\upsilon}^{s}_{\mathsf{abs}}(F(v))$	NSATO6
		$\hat{\hat{\upsilon}}^{s}_{\mathsf{abs}}(\int_{v \in S} F(v)) = \int_{v \in S} \hat{\hat{\upsilon}}^{s}_{\mathsf{abs}}(F(v))$	NSAI6
		$\sigma^{u}_{\mathsf{abs}}(x) \mathbin{\|\!\lfloor} \int_{\epsilon \in \mathbb{I}} \hat{\upsilon}^{\epsilon}_{\mathsf{abs}}(y) = \int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{\delta})$	NSACM0

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. Except for axioms INT7NSAa and INT8NSAa, the remaining axioms are easily understood by realizing that integration is a form of alternative composition over a set of alternatives. Axioms INT10NSA, INT11, INT12, INT13, INT14, INT15, INT16, NSATO6 and NSAI6 can simply be regarded as variants of axioms NSAT3, A2, A4, CM4, CM8, CM9, D3, NSATO4 and NSAI4, respectively. Axioms INT7NSAa and INT8NSAa reveal an identification made in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$ which needs further explanation. It will be discussed after the presentation of the operational semantics of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$.

Except for axioms INT8NSAa, INT8NSAb, INT9NSA, NSAINT, NSACM0, the axioms for integration in $\mathrm{ACP}^{\mathrm{nsat}}$ are simple reformulations of axioms for integration in $\mathrm{ACP}^{\mathrm{sat}}$. Axioms INT8NSAa, INT8NSAb and INT9NSA are closely related to axioms INT8SAa, INT8SAb and INT9SA of $\mathrm{ACP}^{\mathrm{sat}}$ anyway, as will become apparent in Section 3.2. Actually, the choice of axioms INT8NSAa, INT8NSAb, INT9NSA, NSAINT and NSACM0 is rather ad hoc: they are useful, cannot be derived from the other axioms, and hold in the model described below.

The following example illustrates, by means of simple calculations using the axioms of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$, what can be done using nonstandard real numbers in process algebra with nonstandard timing. An additional example will be given at the end of Section 3.2.

Example 2.1. Let $r \in \mathbb{R}$ and $r > 0$. Then, by the definition of $\mathbb{I}$, we have that $r > \epsilon$ for all $\epsilon \in \mathbb{I}$. This implies that $r - \epsilon > 0$ and $r = \epsilon + (r - \epsilon)$ for all $\epsilon \in \mathbb{I}$. Hence, we can easily derive

Table 6. Additional rules for ACP$^{\mathrm{nsat}}$I ($a \in \mathsf{A}$, $s, s', t \geq 0$, $u, u' > 0$)

$$\frac{\langle F(t), s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \int_{v \in S} F(v), s\rangle \xrightarrow{a} \langle x', s\rangle}\ t \in S \qquad \frac{\langle F(t), s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle \int_{v \in S} F(v), s\rangle \xrightarrow{a} \langle \surd, s\rangle}\ t \in S$$

$$\frac{\langle F(t), s\rangle \overset{u}{\longmapsto} \langle F(q), s+u\rangle}{\langle \int_{v \in S} F(v), s\rangle \overset{u}{\longmapsto} \langle \int_{v \in S} F(v), s+u\rangle}\ t \in S$$

$$\frac{\{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle, \langle x, s+u\rangle \overset{a}{\not\to} \mid u < s' + u', a \in \mathsf{A}\}}{\langle x, s+s'\rangle \overset{u'}{\longmapsto} \langle x, s+s'+u'\rangle}$$

$$\frac{\{\langle x, s\rangle \overset{u}{\longmapsto} \langle x, s+u\rangle, \langle x, s+u\rangle \overset{a}{\not\to} \mid u < s' + u', a \in \mathsf{A}\}}{\langle x, s'\rangle \overset{s+u'}{\longmapsto} \langle x, s+s'+u'\rangle}$$

from the axioms of ACP$^{\mathrm{nsat}}$I that $\sigma^r_{\mathsf{abs}}(x) = \int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\sigma^{r-\epsilon}_{\mathsf{abs}}(x))$. Using this, we can, for example, also derive that $\sigma^r_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\hat{a})) + \int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\hat{\delta}) = \int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\sigma^{r-\epsilon}_{\mathsf{abs}}(\int_{\epsilon' \in \mathbb{I}} \sigma^{\epsilon'}_{\mathsf{abs}}(\hat{a})) + \hat{\delta}) = \sigma^r_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\hat{a}))$. In Section 3.2, we will see that the terms $\sigma^r_{\mathsf{abs}}(\tilde{a}) + \tilde{\delta}$ and $\sigma^r_{\mathsf{abs}}(\tilde{a})$ of ACP$^{\mathrm{sat}}$I, in which only standard real numbers are allowed, can be viewed as abbreviations of the terms $\sigma^r_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\hat{a})) + \int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\hat{\delta})$ and $\sigma^r_{\mathsf{abs}}(\int_{\epsilon \in \mathbb{I}} \sigma^\epsilon_{\mathsf{abs}}(\hat{a}))$ of ACP$^{\mathrm{nsat}}$I.

Semantics of ACP$^{\mathrm{nsat}}$I

The structural operational semantics of ACP$^{\mathrm{nsat}}$I is described by the rules for ACP$^{\mathrm{nsat}}$ and the rules given in Table 6. We write $\langle t, s\rangle \overset{a}{\not\to}$ for the set consisting of all transition formulas $\neg(\langle t, s\rangle \xrightarrow{a} \langle t', s\rangle)$ where t' is a closed term of ACP$^{\mathrm{nsat}}$I and the transition formula $\neg(\langle t, s\rangle \xrightarrow{a} \langle \surd, s\rangle)$. If an idling process becomes inactive at a point of time s, idling must in any case have taken place up to s. In the presence of integration, one could make a distinction between the case that s is included in the idling period and the case that s is not included in the idling period. Definitely, identification of the two cases is a useful abstraction. We have achieved the identification by means of the last two rules in Table 6. These rules were not introduced earlier because it is entirely due to integration that the cases can be distinguished. The option to leave out the last two rules in Table 6 is plausible as well. The only consequence of that choice is that axioms INT7NSAa and INT8NSAa must be removed from the axiom system.

A bisimulation model of the axioms of ACP$^{\mathrm{nsat}}$ extended with integration can be constructed in the same way as for ACP$^{\mathrm{nsat}}$.

2.5. Initial Abstraction

In order to cover processes of which the timing depends on their initialization time, we add nonstandard initial abstraction to ACP$^{\mathrm{nsat}}$I. Such processes, called processes with parametric timing, can be perceived as functions that map each $s \in \mathbb{F}$ to an ordinary process initialized at point of time s. Nonstandard initial abstraction is a mechanism to form such functions. It is represented by the variable-binding operator $\surd$. Let P be an expression, possibly containing variable v, such that $P[s/v]$ represents a process for all $s \in \mathbb{F}$. Then the *nonstandard initial abstraction* $\surd v \mathbin{.} P$ is a process that, when initialized at point of time s, proceeds as $P[s/v]$. That is, $\hat{\overline{\upsilon}}^s_{\mathsf{abs}}(\surd v \mathbin{.} P)$ behaves the same as $\hat{\overline{\upsilon}}^s_{\mathsf{abs}}(P[s/v])$. Nonstandard initial abstraction is the primary way of forming processes with parametric timing. The ways of combining

Table 7. Additional axioms for $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ ($s \geq 0$)

$\surd v \,.\, F(v) = \surd w \,.\, F(w)$	NSIA1	$\hat{v}^{s}_{\mathsf{abs}}(\surd v \,.\, F(v)) = \surd v \,.\, \hat{v}^{s}_{\mathsf{abs}}(F(v))$	NSIA10
$\hat{\overline{v}}^{s}_{\mathsf{abs}}(\surd v \,.\, F(v)) = \hat{\overline{v}}^{s}_{\mathsf{abs}}(F(s))$	NSIA2	$(\surd v \,.\, F(v)) \mathbin{\lfloor\!\lfloor} x = \surd v \,.\, (F(v) \mathbin{\lfloor\!\lfloor} \hat{\overline{v}}^{v}_{\mathsf{abs}}(x))$	NSIA11
$\surd v \,.\, (\surd w \,.\, K(v,w)) = \surd v \,.\, K(v,v)$	NSIA3	$x \mathbin{\lfloor\!\lfloor} (\surd v \,.\, F(v)) = \surd v \,.\, (\hat{\overline{v}}^{v}_{\mathsf{abs}}(x) \mathbin{\lfloor\!\lfloor} F(v))$	NSIA12
$x = \surd v \,.\, x$	NSIA4	$(\surd v \,.\, F(v)) \mid x = \surd v \,.\, (F(v) \mid \hat{\overline{v}}^{v}_{\mathsf{abs}}(x))$	NSIA13
$(\forall v \in \mathbb{F} \bullet \hat{\overline{v}}^{v}_{\mathsf{abs}}(x) = \hat{\overline{v}}^{v}_{\mathsf{abs}}(y)) \Rightarrow x = y$	NSIA5	$x \mid (\surd v \,.\, F(v)) = \surd v \,.\, (\hat{\overline{v}}^{v}_{\mathsf{abs}}(x) \mid F(v))$	NSIA14
$\sigma^{s}_{\mathsf{abs}}(\hat{a}) \cdot x = \sigma^{s}_{\mathsf{abs}}(\hat{a}) \cdot \hat{\overline{v}}^{s}_{\mathsf{abs}}(x)$	NSIA6	$\partial_H(\surd v \,.\, F(v)) = \surd v \,.\, \partial_H(F(v))$	NSIA15
$\sigma^{s}_{\mathsf{abs}}(\surd v \,.\, F(v)) = \sigma^{s}_{\mathsf{abs}}(F(0))$	NSIA7	$\int_{v \in S}(\surd w \,.\, K(v,w)) =$	
$(\surd v \,.\, F(v)) + x = \surd v \,.\, (F(v) + \hat{\overline{v}}^{v}_{\mathsf{abs}}(x))$	NSIA8	$\surd w \,.\, (\int_{v \in S} K(v,w))$ if $v \neq w$	NSIA17
$(\surd v \,.\, F(v)) \cdot x = \surd v \,.\, (F(v) \cdot x)$	NSIA9		

and timing processes available in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$ can simply be lifted to processes with parametric timing. For example, the process $\hat{\overline{v}}^{2}_{\mathsf{abs}}((\surd s \,.\, \sigma^{s+3}_{\mathsf{abs}}(\hat{a})) + (\surd s \,.\, \sigma^{s+1}_{\mathsf{abs}}(\hat{b})))$ behaves the same as $\sigma^{5}_{\mathsf{abs}}(\hat{a}) + \sigma^{3}_{\mathsf{abs}}(\hat{b})$.

Axioms of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$

The axiom system of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ consists of the axioms of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$ and the equations given in Table 7. We use F and G as in Table 5, and K similarly. Axioms NSIA1 and NSIA2 are similar to the α- and β-conversion rules of λ-calculus. Axiom NSIA3 shows that multiple nonstandard initial abstractions can be replaced by one. Axiom NSIA4 points out that ordinary processes are special cases of processes with parametric timing: their behaviour do not vary with different initialization times. Axiom NSIA5 is an extensionality axiom. Axiom NSIA6 expresses that in the case where 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 $\mathrm{ACP}^{\mathrm{nsat}}$ to processes with parametric timing.

The axioms for nonstandard initial abstraction in $\mathrm{ACP}^{\mathrm{nsat}}$ are simple reformulations of the axioms for standard initial abstraction in $\mathrm{ACP}^{\mathrm{sat}}$.

Semantics of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$

To obtain a model of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$, we extend the bisimulation model $\mathcal{M}$ of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$ to

$$\mathcal{M}^* = \{f : \mathbb{F} \to \mathcal{M} \mid \forall v \in \mathbb{F} \bullet f(v) = \hat{\overline{v}}^{v}_{\mathsf{abs}}(f(v))\}$$

and define the constants and operators of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ on $\mathcal{M}^*$ as in Table 8. The auxiliary function $* : \mathcal{M} \times \mathcal{M}^* \to \mathcal{M}$, used in the definition of $\cdot$ on $\mathcal{M}^*$, is defined in Table 9. We use $p, q, \ldots$ to denote elements of $\mathcal{M}$, $f, g, \ldots$ to denote elements of $\mathcal{M}^*$, ψ to denote elements of $\mathbb{F} \to \mathcal{M}$, and ϕ to denote elements of $\mathbb{F} \to \mathcal{M}^*$. We use λ-notation for functions.

Table 8. Definition of operators of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}$ on $\mathcal{M}^*$

$\hat{a} = \lambda w \,.\, \hat{\overline{\upsilon}}^{w}_{\mathsf{abs}}(\hat{a})$ for each $a \in \mathsf{A}_\delta$	$f \parallel g = \lambda w \,.\, (f(w) \parallel g(w))$
$\sigma^{v}_{\mathsf{abs}}(f) = \lambda w \,.\, \hat{\overline{\upsilon}}^{w}_{\mathsf{abs}}(\sigma^{v}_{\mathsf{abs}}(f(0)))$	$f \mathbin{\|\!\_} g = \lambda w \,.\, (f(w) \mathbin{\|\!\_} g(w))$
$f + g = \lambda w \,.\, (f(w) + g(w))$	$f \mid g = \lambda w \,.\, (f(w) \mid g(w))$
$f \cdot g = \lambda w \,.\, (f(w) * g)$	$\partial_H(f) = \lambda w \,.\, \partial_H(f(w))$
$\hat{\upsilon}^{v}_{\mathsf{abs}}(f) = \lambda w \,.\, \hat{\overline{\upsilon}}^{w}_{\mathsf{abs}}(\hat{\upsilon}^{v}_{\mathsf{abs}}(f(w)))$	$\int(S, \varphi) = \lambda w \,.\, \int(S, \lambda w' \,.\, \varphi(w')(w))$
$\hat{\overline{\upsilon}}^{v}_{\mathsf{abs}}(f) = f(v)$	$\surd(\varphi) = \lambda w \,.\, \hat{\overline{\upsilon}}^{w}_{\mathsf{abs}}(\varphi(w))$

Table 9. Definition of $*$

$\hat{a} * f = \hat{a} \cdot f(0)$ for each $a \in \mathsf{A}_\delta$
$\sigma^{v}_{\mathsf{abs}}(p) * f = \sigma^{v}_{\mathsf{abs}}(p * \lambda w \,.\, f(v + w))$
$(p + q) * f = p * f + q * f$
$(p \cdot q) * f = p \cdot (q * f)$
$\int(S, \psi) * f = \int(S, \lambda w \,.\, (\psi(w) * f))$

3. Embedding of Process Algebra with Standard Timing

In this section, we show that $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ generalizes $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ by giving an embedding. 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. We characterize the embedding by explicit definitions of the constants and operators of the source process algebra that are not available in the target process algebra. For more information on this approach, see [6, 7].

First, we shortly review $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$. This is done by giving the main similarities and dissimilarities with $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. After that, we give the embedding and elaborate on it. For reference, the axioms of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are given in Appendix A. For an extensive presentation of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$, see [6, 7].

3.1. Process Algebra with Standard Timing

Basic Process Algebra

Like in the version of ACP with timing presented in Section 2, the atomic processes are undelayable actions. The undelayable action a ($a \in \mathsf{A}$) is now written $\tilde{a}$. We also have again absolute delay as the basic way of timing processes, alternative composition and sequential composition as the basic ways of combining processes, and undelayable deadlock to deal with processes that become inactive. The undelayable deadlock is now written $\tilde{\tilde{\delta}}$. In addition, we have standard absolute time-out and standard absolute initialization. The standard absolute time-out and standard absolute initialization of process p at point of time r ($r \in \mathbb{R}^{\geq}$) are written $\upsilon^{r}_{\mathsf{abs}}(p)$ and $\overline{\upsilon}^{r}_{\mathsf{abs}}(p)$, respectively. Unlike in the version of ACP with timing presented in Section 2, we have the *deadlocked process*, written $\dot{\delta}$ which can be viewed as a trace of a process that has become inactive before point of time 0. Thus, $\sigma^{3}_{\mathsf{abs}}(\dot{\delta})$ is the process that idles from point of time 0 till arbitrarily close to point of time 3 without ever becoming able to perform an action or to idle further.

Inclusion of the possibility of two or more actions to be performed consecutively at the same point in time, has several consequences. First of all, the deadlocked process is needed to handle situations in which processes exhibit inconsistent timing. Secondly, sequential composition differs in detail: the second process may now perform its first action also at the point of time that the first process terminates successfully. Absolute time-out and absolute initialization differ in detail as well. This is because the differences concerned make the operators more appropriate in the presence of the possibility of two or more actions to be performed consecutively at the same point in time.

The axiom system of $\mathrm{BPA}^{\mathrm{sat}}$ consists of the equations given in Table 14 (Appendix A). The differences between corresponding axioms of $\mathrm{BPA}^{\mathrm{sat}}$ and $\mathrm{BPA}^{\mathrm{nsat}}$ were already discussed in Section 2.2. In addition, some axioms of $\mathrm{BPA}^{\mathrm{sat}}$ have no corresponding axioms in $\mathrm{BPA}^{\mathrm{nsat}}$ due to the presence of both undelayable deadlock and the deadlocked process.

Algebra of Communicating Processes

Like in the version of ACP with timing presented in Section 2, we have parallel composition and encapsulation to capture parallelism and communication. In addition, we have again the two well-known variants of parallel composition: left merge and communication merge. Unlike in the version of ACP with timing presented in Section 2, standard absolute time-out cannot be used to keep processes entirely from idling. For that purpose, we have still another auxiliary operator: absolute undelayable time-out. The *absolute undelayable time-out* of process p, written $\nu_{\mathsf{abs}}(p)$, behaves like the part of p that starts performing actions at point of time 0 if p is capable of doing so, and otherwise like undelayable deadlock. For example, $\nu_{\mathsf{abs}}(\sigma^3_{\mathsf{abs}}(\tilde{a}) + \tilde{b})$ behaves the same as $\tilde{b}$.

Inclusion of the possibility of two or more actions to be performed consecutively at the same point in time, has an additional consequence in the presence of parallelism. Parallel composition differs in detail: it may now also start with performing an action of one of the processes at the ultimate time for the other process to start performing actions or to become inactive.

The additional axioms for $\mathrm{ACP}^{\mathrm{sat}}$ are the equations given in Table 15 (Appendix A). The differences between corresponding additional axioms for $\mathrm{ACP}^{\mathrm{sat}}$ and $\mathrm{ACP}^{\mathrm{nsat}}$ were already discussed in Section 2.3. Some additional axioms of $\mathrm{ACP}^{\mathrm{sat}}$ have no corresponding additional axioms in $\mathrm{ACP}^{\mathrm{nsat}}$. This is again due to the presence of both undelayable deadlock and the deadlocked process.

Integration and Initial Abstraction

Integration is essentially the same as in the version of ACP with timing presented in Section 2. The addition of integration to $\mathrm{ACP}^{\mathrm{sat}}$ does not go hand in hand with the identification discussed at the end of Section 2.4 because the distinction between undelayable deadlock and the deadlocked process would be blurred. The axioms for integration are the equations given in Table 16 (Appendix A). The differences between corresponding axioms for integration in $\mathrm{ACP}^{\mathrm{sat}}$ and $\mathrm{ACP}^{\mathrm{nsat}}$ were already discussed in Section 2.4.

Initial abstraction is essentially the same as in the version of ACP with timing presented in Section 2. The axioms for standard initial abstraction are the equations given in Table 17 (Appendix A).

Table 10. Explicit definition constants/operators of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$

$\dot{\delta} = \hat{\delta}$	
$\tilde{a} = \int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{a})$	for each $a \in \mathsf{A}$
$\tilde{\delta} = \int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{\delta})$	
$\sigma^{p}_{\mathsf{abs}}(x) = \sigma^{p}_{\mathsf{abs}}(x)$	
$\upsilon^{p}_{\mathsf{abs}}(x) = \hat{\upsilon}^{p}_{\mathsf{abs}}(x)$	
$\overline{\upsilon}^{p}_{\mathsf{abs}}(x) = \hat{\overline{\upsilon}}^{p}_{\mathsf{abs}}(x)$	
$\nu_{\mathsf{abs}}(x) = \int_{\epsilon \in \mathbb{I}} \hat{\upsilon}^{\epsilon}_{\mathsf{abs}}(x)$	
$\int_{v \in V} F(v) = \int_{v \in V} F(v)$	
$\surd_{\mathsf{s}} v \,.\, F(v) = \surd v \,.\, F(v^{\circ})$	

3.2. Embedding

The undelayable action constants $\tilde{a}$ ($a \in \mathsf{A}$), the undelayable deadlock constant $\tilde{\delta}$, the deadlocked process constant $\dot{\delta}$, the standard absolute time-out operator υ_{abs}, the standard absolute initialization operator $\overline{\upsilon}_{\mathsf{abs}}$, the absolute undelayable time-out operator ν_{abs}, and the standard initial abstraction operator $\surd_{\mathsf{s}}$ of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are not present in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. Besides, the operator σ_{abs} has an element of $\mathbb{R}^{\geq}$ instead of an element of $\mathbb{F}$ as its first argument and the operator $\int$ has a subset of $\mathbb{R}^{\geq}$ instead of a subset of $\mathbb{F}$ as its first argument. Explicit definitions of these constants and operators in $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are given in Table 10. Notice that the explicit definitions of the operators σ_{abs} and $\int$ express that they are restrictions of the corresponding operators of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. It is rather straighforward to prove the following theorem.

Theorem 3.1. The explicit definitions given in Table 10 induce an embedding of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$.

Proof:
In order to prove that the explicit definitions given in Table 10 induce an embedding of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$, we have to check that the axioms of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are derivable for closed terms from the axioms of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ and the explicit definitions. Using Lemmas 3.1, 3.2 and 3.3 given below, it is straightforward to check the derivability of the axioms of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$. □

The following three lemmas state results, concerning $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$-terms that correspond to $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$-terms, that are used in the proof of Theorem 3.1.

Lemma 3.1. For all closed $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$-terms t that correspond to $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$-terms, either $t = \hat{\delta}$ is derivable, or there exists an $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$-term t' such that $t = \int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(t')$ is derivable, or there exists an $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$-term t' such that $t = \surd v \,.\, t'[v^{\circ}/v]$ is derivable and $t'[v^{\circ}/v]$ corresponds to an $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$-term.

Proof:
Straightforward by induction on the structure of t. □

Lemma 3.2. For all closed $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$-terms t that correspond to $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$-terms, $\hat{\overline{\upsilon}}^{0}_{\mathsf{abs}}(t) = \sigma^{0}_{\mathsf{abs}}(t)$. Besides, $\hat{\overline{\upsilon}}^{0}_{\mathsf{abs}}(t) = t$ if t corresponds to an $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}$-term.

Table 11. Axioms for standardization ($a \in \mathsf{A}_\delta$, $s \geq 0$)

$\mathcal{S}(\hat{a}) = \tilde{a}$
$\mathcal{S}(\sigma^{s}_{\mathsf{abs}}(x)) = \sigma^{s^\circ}_{\mathsf{abs}}(\mathcal{S}(x))$
$\mathcal{S}(x + y) = \mathcal{S}(x) + \mathcal{S}(y)$
$\mathcal{S}(x \cdot y) = \mathcal{S}(x) \cdot \mathcal{S}(y)$
$\mathcal{S}(\int_{v \in S} F(v)) = \int_{v \in S} \mathcal{S}(F(v))$
$\mathcal{S}(\surd v \, . \, F(v)) = \surd_{\mathsf{s}} v \, . \, \mathcal{S}(F(v))$

Proof:
Easy by case distinction according to Lemma 3.1. □

Lemma 3.3. All closed ACP$^{\mathrm{nsat}}$I$\surd$-terms t that correspond to ACP$^{\mathrm{sat}}$I$\surd$-terms are *standardly initialized*, i.e. $t = \surd v \, . \, \hat{\overline{v}}^{v^\circ}_{\mathsf{abs}}(t)$.

Proof:
Straightforward by case distinction according to Lemma 3.1, using the extensionality axiom NSIA5. □

Returning to the proof of Theorem 3.1, it is worth noticing that axioms INT8SAa, INT8SAb and INT9SA can simply be derived by first applying the relevant explicit definitions and then INT8NSAa, INT8NSAb and INT9NSA, respectively. Indeed, the latter axioms are indispensable in the sense that without these axioms, or equivalent ones, the explicit definitions would not induce an embedding.

The embedding concerned corresponds to the view that, for a standard time process, the execution of its actions is always timed with respect to the clusters of nonstandard points of time infinitely close to a standard point of time. The embedding shows an interesting viewpoint on the difference between the deadlocked process and undelayable deadlock in ACP$^{\mathrm{sat}}$: the deadlocked process cannot idle at all, but undelayable deadlock can still idle for an infinitely small period of time. Thus, in the case of undelayable deadlock, a parallel process may still perform some actions whereas this is impossible in the case of the deadlocked process.

We will elaborate on the embedding. We will define the notion of a standardized process in terms of the auxiliary *standardization* operator $\mathcal{S}$ of which the defining axioms are given in Table 11. The transition rules for standardization on processes with nonstandard absolute timing are given in Table 12. In Table 13, standardization is defined on processes with nonstandard parametric timing. The definition shows that standardization shifts the capabilities of a process at any point of time s to the cluster of nonstandard points of time infinitely close to the standard part of s. Notice that, consequently, processes may lose deadlock possibilities by standardization. For example, if $\epsilon \in \mathbb{I}$, $\mathcal{S}(\sigma^{1+\epsilon}_{\mathsf{abs}}(\hat{a}) + \sigma^{1+2\epsilon}_{\mathsf{abs}}(\hat{\delta})) = \sigma^{1}_{\mathsf{abs}}(\tilde{a}) + \sigma^{1}_{\mathsf{abs}}(\tilde{\delta}) = \sigma^{1}_{\mathsf{abs}}(\tilde{a})$. However, such processes are not standardized.

A process with nonstandard parametric timing x is *standardized* if $x = \mathcal{S}(x)$. For any such process, the following holds: if an action can be performed at some nonstandard point of time s, it can also be performed at any other nonstandard point of time s' such that $s \approx s'$. A standardized process is never capable of performing an action at a standard point of time. Notice that standardization is a retraction, i.e. we have that $\mathcal{S}(\mathcal{S}(x)) = \mathcal{S}(x)$. Notice further that $\mathcal{S}(\hat{\delta}) \neq \hat{\delta}$ and $\dot{\hat{\delta}} = \hat{\delta}$. Hence, $\dot{\hat{\delta}}$ is not standardized. A process with nonstandard parametric timing x is *weakly standardized* if x is standardized or $x = \hat{\delta}$.

Table 12. Rules for standardization ($a \in \mathsf{A}$, $s, s' \geq 0$, $u, u' > 0$)

$$\frac{\langle x, s\rangle \xrightarrow{a} \langle x', s\rangle}{\langle \mathcal{S}(x), s'\rangle \xrightarrow{a} \langle \mathcal{S}(x'), s'\rangle}\; s' \in \{s^\circ + \epsilon \mid \epsilon \in \mathbb{I}\} \qquad \frac{\langle x, s\rangle \xrightarrow{a} \langle \surd, s\rangle}{\langle \mathcal{S}(x), s'\rangle \xrightarrow{a} \langle \surd, s'\rangle}\; s' \in \{s^\circ + \epsilon \mid \epsilon \in \mathbb{I}\}$$

$$\frac{\langle x, s\rangle \overset{u}{\longmapsto} \langle x', s + u\rangle}{\langle \mathcal{S}(x), s\rangle \overset{u'}{\longmapsto} \langle \mathcal{S}(x), s + u'\rangle}\; s + u' \in \{s + u^\circ + \epsilon \mid \epsilon \in \mathbb{I} \cup \{0\}\},\, s + u \leq s + u'$$

$$\frac{\langle x, 0\rangle \overset{u}{\longmapsto} \langle x', u\rangle}{\langle \mathcal{S}(x), u\rangle \overset{u'}{\longmapsto} \langle \mathcal{S}(x), u + u'\rangle}\; u + u' \in \{u^\circ + \epsilon \mid \epsilon \in \mathbb{I} \cup \{0\}\} \qquad \frac{}{\langle \mathcal{S}(x), 0\rangle \overset{\epsilon}{\longmapsto} \langle \mathcal{S}(x), \epsilon\rangle}\; \epsilon \in \mathbb{I}$$

Table 13. Definition of standardization on $\mathcal{M}^*$

$$\mathcal{S}(f) = \lambda t \,.\, \mathcal{S}(f(t))$$

The set of weakly standardized processes includes the embedded constants of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ and is closed under the embedded operators of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$. This suggests the construction of a model of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$. We claim that this model is isomorphic to the model of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ presented in [6].

The following example illustrates, by means of simple calculations, what can be done with the standardization operator.

Example 3.1. Using $\epsilon^\circ = 0$ for all $\epsilon \in \mathbb{I}$, we calculate $\mathcal{S}(\tilde{a}) = \mathcal{S}(\int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon}_{\mathsf{abs}}(\hat{a})) = \int_{\epsilon \in \mathbb{I}} \mathcal{S}(\sigma^{\epsilon}_{\mathsf{abs}}(\hat{a})) = \int_{\epsilon \in \mathbb{I}} \sigma^{\epsilon^\circ}_{\mathsf{abs}}(\mathcal{S}(\hat{a})) = \int_{\epsilon \in \mathbb{I}} \sigma^{0}_{\mathsf{abs}}(\tilde{a}) = \tilde{a}$. Hence, using $p^\circ = p$ for all $p \in \mathbb{R}^{\geq}$, we calculate, for an arbitrary $p \in \mathbb{R}^{\geq}$, $\mathcal{S}(\sigma^{p}_{\mathsf{abs}}(\tilde{a})) = \sigma^{p^\circ}_{\mathsf{abs}}(\mathcal{S}(\tilde{a})) = \sigma^{p}_{\mathsf{abs}}(\tilde{a})$. Using further properties concerning standard parts, we also calculate, for an arbitrary $\epsilon \in \mathbb{I}$, $\mathcal{S}(\sigma^{1+\epsilon}_{\mathsf{abs}}(\hat{a}) \parallel \sigma^{1+2\epsilon}_{\mathsf{abs}}(\hat{b})) = \mathcal{S}(\sigma^{1+\epsilon}_{\mathsf{abs}}(\hat{a}) \cdot \sigma^{1+2\epsilon}_{\mathsf{abs}}(\hat{b})) = \mathcal{S}(\sigma^{1+\epsilon}_{\mathsf{abs}}(\hat{a})) \cdot \mathcal{S}(\sigma^{1+2\epsilon}_{\mathsf{abs}}(\hat{b})) = \sigma^{(1+\epsilon)^\circ}_{\mathsf{abs}}(\mathcal{S}(\hat{a})) \cdot \sigma^{(1+2\epsilon)^\circ}_{\mathsf{abs}}(\mathcal{S}(\hat{b})) = \sigma^{1}_{\mathsf{abs}}(\tilde{a}) \cdot \sigma^{1}_{\mathsf{abs}}(\tilde{b})$. Notice that $\mathcal{S}(\sigma^{1+\epsilon}_{\mathsf{abs}}(\hat{a}) \parallel \sigma^{1+2\epsilon}_{\mathsf{abs}}(\hat{b})) \neq \mathcal{S}(\sigma^{1+\epsilon}_{\mathsf{abs}}(\hat{a})) \parallel \mathcal{S}(\sigma^{1+2\epsilon}_{\mathsf{abs}}(\hat{b})) = \sigma^{1}_{\mathsf{abs}}(\tilde{a}) \parallel \sigma^{1}_{\mathsf{abs}}(\tilde{b})$.

It is straightforward to show that all standardized processes are standardly initialized (cf. Lemma 3.3). This means that for standardized processes, the initialization time can always be taken to be a standard point of time. However, not all standardly initialized processes are standardized, e.g. $\surd v \,.\, \sigma^{v^\circ}_{\mathsf{abs}}(\hat{a})$ is standardly initialized but not standardized.

4. Concluding Remarks

A new version of ACP with timing, called $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$, has been presented that, unlike the versions presented in [6], excludes the possibility of two or more actions to be performed consecutively at the same point in time. Because it includes nonstandard non-negative real numbers in the time domain, 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. This is substantiated by showing, taking this view as starting-point, that there exists an embedding of the most general version of ACP with timing from [6], called $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$, in this new version.

The constants and operators of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ and the ones of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are very much alike. As a result, the essential differences are clearly reflected by differences in the axiom systems. The embedding of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ shows that, despite those differences, these versions of ACP with timing are yet fairly similar. In addition to that, the embedding presents new insight into the difference between the deadlocked process and undelayable deadlock in $\mathrm{ACP}^{\mathrm{sat}}$.

Unlike in [4], the choice of the non-negative finite numbers from a nonstandard model of the real numbers as time domain has been made a priori in this paper. In the case where the choice had not been made a priori, it would have been virtually impossible to introduce axioms INT8NSAa, INT8NSAb, INT9NSA and NSAINT, or equivalent axioms. Without such axioms, it can only be shown that there exists an embedding of $\mathrm{ACP}^{\mathrm{sat}}\surd$, i.e. $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ without integration, in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. This makes also clear why in [4] the integration operator is absent from the embedded process algebra.

No applications of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ have been given in this paper. Slight adaptations of the applications of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ given in [7] can be transferred to $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$. The required adaptations concern the exclusion of the possibility of two or more actions to be performed consecutively at the same point in time. In order to achieve that entirely independent actions can be performed at the same point in time, the multi-actions of [1] could be used. This is, however, a real burden when describing and analyzing systems in which actions occur that are entirely independent. That is one of the primary reasons why $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ was devised in [6]. The usefulness of $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ for systems in which no independent actions occur, has not yet been investigated.

In any case, the primary motivation to introduce $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ was to get an important issue relevant to dealing with time-dependent behaviour of processes, viz. whether the possibility of two or more actions to be performed consecutively at the same point in time should be excluded or not, conceptually clear in the same setting as the issues studied in [6]. Viewed in that light, the embedding of $\mathrm{ACP}^{\mathrm{sat}}\surd$ in $\mathrm{ACP}^{\mathrm{nsat}}\mathrm{I}\surd$ is highly relevant. It provides a justification of the choice not to exclude this possibility in $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$.

A. Axioms of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$

For reference, the axioms of $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ are given in this appendix. The axiom system of $\mathrm{BPA}^{\mathrm{sat}}$ consists of the equations given in Table 14. The axiom system of $\mathrm{ACP}^{\mathrm{sat}}$ consists of the equations given in Tables 14 and 15. The additional axioms for integration and standard initial abstraction are given in Tables 16 and 17, respectively.

Table 14. Axioms of $\mathrm{BPA}^{\mathrm{sat}}$ ($a \in \mathsf{A}_\delta$, $p, q \geq 0$, $r > 0$)

$x + y = y + x$	A1	$\sigma^0_{\mathsf{abs}}(x) = \overline{\upsilon}^0_{\mathsf{abs}}(x)$	SAT1
$(x + y) + z = x + (y + z)$	A2	$\sigma^p_{\mathsf{abs}}(\sigma^q_{\mathsf{abs}}(x)) = \sigma^{p+q}_{\mathsf{abs}}(x)$	SAT2
$x + x = x$	A3	$\sigma^p_{\mathsf{abs}}(x) + \sigma^p_{\mathsf{abs}}(y) = \sigma^p_{\mathsf{abs}}(x + y)$	SAT3
$(x + y) \cdot z = (x \cdot z) + (y \cdot z)$	A4	$\sigma^p_{\mathsf{abs}}(x) \cdot \upsilon^p_{\mathsf{abs}}(y) = \sigma^p_{\mathsf{abs}}(x \cdot \dot{\delta})$	SAT4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5	$\sigma^p_{\mathsf{abs}}(x) \cdot (\upsilon^p_{\mathsf{abs}}(y) + \sigma^p_{\mathsf{abs}}(z)) = \sigma^p_{\mathsf{abs}}(x \cdot \overline{\upsilon}^0_{\mathsf{abs}}(z))$	SAT5
$x + \dot{\delta} = x$	A6ID	$\sigma^p_{\mathsf{abs}}(\dot{\delta}) \cdot x = \sigma^p_{\mathsf{abs}}(\dot{\delta})$	SAT6
$\dot{\delta} \cdot x = \dot{\delta}$	A7ID	$\tilde{a} + \tilde{\delta} = \tilde{a}$	A6SAa
		$\sigma^r_{\mathsf{abs}}(x) + \tilde{\delta} = \sigma^r_{\mathsf{abs}}(x)$	A6SAb
		$\tilde{\delta} \cdot x = \tilde{\delta}$	A7SA
		$\overline{\upsilon}^0_{\mathsf{abs}}(\dot{\delta}) = \dot{\delta}$	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 15. Additional axioms for $\mathrm{ACP}^{\mathrm{sat}}$ ($a, b \in \mathsf{A}_\delta$, $c \in \mathsf{A}$, $p \geq 0$, $r > 0$)

$x \parallel y = (x \mathbin{⌊\!⌊} y + y \mathbin{⌊\!⌊} x) + x \mid y$	CM1	$\tilde{a} \cdot x \mid \tilde{b} = (\tilde{a} \mid \tilde{b}) \cdot x$	CM5SA
$\dot{\delta} \mathbin{⌊\!⌊} x = \dot{\delta}$	CMID1	$\tilde{a} \mid \tilde{b} \cdot x = (\tilde{a} \mid \tilde{b}) \cdot x$	CM6SA
$x \mathbin{⌊\!⌊} \dot{\delta} = \dot{\delta}$	CMID2	$\tilde{a} \cdot x \mid \tilde{b} \cdot y = (\tilde{a} \mid \tilde{b}) \cdot (x \parallel y)$	CM7SA
$\tilde{a} \mathbin{⌊\!⌊} (x + \tilde{\delta}) = \tilde{a} \cdot (x + \tilde{\delta})$	CM2SA	$(\nu_{\mathsf{abs}}(x) + \tilde{\delta}) \mid \sigma^r_{\mathsf{abs}}(y) = \tilde{\delta}$	SACM3
$\tilde{a} \cdot x \mathbin{⌊\!⌊} (y + \tilde{\delta}) = \tilde{a} \cdot (x \parallel (y + \tilde{\delta}))$	CM3SA	$\sigma^r_{\mathsf{abs}}(x) \mid (\nu_{\mathsf{abs}}(y) + \tilde{\delta}) = \tilde{\delta}$	SACM4
$\sigma^r_{\mathsf{abs}}(x) \mathbin{⌊\!⌊} (\nu_{\mathsf{abs}}(y) + \tilde{\delta}) = \tilde{\delta}$	SACM1	$\sigma^p_{\mathsf{abs}}(x) \mid \sigma^p_{\mathsf{abs}}(y) = \sigma^p_{\mathsf{abs}}(x \mid y)$	SACM5
$\sigma^p_{\mathsf{abs}}(x) \mathbin{⌊\!⌊} (\upsilon^p_{\mathsf{abs}}(y) + \sigma^p_{\mathsf{abs}}(z)) = \sigma^p_{\mathsf{abs}}(x \mathbin{⌊\!⌊} z)$	SACM2	$(x + y) \mid z = x \mid z + y \mid z$	CM8
$(x + y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} z + y \mathbin{⌊\!⌊} z$	CM4	$x \mid (y + z) = x \mid y + x \mid z$	CM9
$\dot{\delta} \mid x = \dot{\delta}$	CMID3		
$x \mid \dot{\delta} = \dot{\delta}$	CMID4	$\tilde{a} \mid \tilde{b} = \tilde{c}$ if $\gamma(a, b) = c$	CF1SA
		$\tilde{a} \mid \tilde{b} = \tilde{\delta}$ if $\gamma(a, b)$ undefined	CF2SA
$\partial_H(\dot{\delta}) = \dot{\delta}$	D0		
$\partial_H(\tilde{a}) = \tilde{a}$ if $a \notin H$	D1SA	$\nu_{\mathsf{abs}}(\dot{\delta}) = \dot{\delta}$	SAU0
$\partial_H(\tilde{a}) = \tilde{\delta}$ if $a \in H$	D2SA	$\nu_{\mathsf{abs}}(\tilde{a}) = \tilde{a}$	SAU1
$\partial_H(\sigma^p_{\mathsf{abs}}(x)) = \sigma^p_{\mathsf{abs}}(\partial_H(x))$	SAD	$\nu_{\mathsf{abs}}(\sigma^r_{\mathsf{abs}}(x)) = \tilde{\delta}$	SAU2
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D3	$\nu_{\mathsf{abs}}(x + y) = \nu_{\mathsf{abs}}(x) + \nu_{\mathsf{abs}}(y)$	SAU3
$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$	D4	$\nu_{\mathsf{abs}}(x \cdot y) = \nu_{\mathsf{abs}}(x) \cdot y$	SAU4

Table 16. Additional axioms for $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}$ ($p \geq 0$)

$\int_{v \in V} F(v) = \int_{w \in V} F(w)$	INT1	$\sup V = p,\ p \in V \Rightarrow$	
$\int_{v \in \emptyset} F(v) = \dot{\delta}$	INT2	$\int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\tilde{\delta}) = \sigma^{p}_{\mathsf{abs}}(\tilde{\delta})$	INT9SA
$\int_{v \in \{p\}} F(v) = F(p)$	INT3	$\int_{v \in V} \sigma^{p}_{\mathsf{abs}}(F(v)) = \sigma^{p}_{\mathsf{abs}}(\int_{v \in V} F(v))$	INT10SA
$\int_{v \in V \cup W} F(v) =$		$\int_{v \in V} (F(v) + G(v)) =$	
$\int_{v \in V} F(v) + \int_{v \in W} F(v)$	INT4	$\int_{v \in V} F(v) + \int_{v \in V} G(v)$	INT11
$V \neq \emptyset \Rightarrow \int_{v \in V} x = x$	INT5	$\int_{v \in V} (F(v) \cdot x) = (\int_{v \in V} F(v)) \cdot x$	INT12
$(\forall v \in V \bullet F(v) = G(v)) \Rightarrow$		$\int_{v \in V} (F(v) \mathbin{\|\!\_} x) = (\int_{v \in V} F(v)) \mathbin{\|\!\_} x$	INT13
$\int_{v \in V} F(v) = \int_{v \in V} G(v)$	INT6	$\int_{v \in V} (F(v) \mid x) = (\int_{v \in V} F(v)) \mid x$	INT14
$\sup V = p \Rightarrow$		$\int_{v \in V} (x \mid F(v)) = x \mid (\int_{v \in V} F(v))$	INT15
$\int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\dot{\delta}) = \sigma^{p}_{\mathsf{abs}}(\dot{\delta})$	INT7SAa	$\int_{v \in V} \partial_H(F(v)) = \partial_H(\int_{v \in V} F(v))$	INT16
V, W unbounded $\Rightarrow$			
$\int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\dot{\delta}) = \int_{v \in W} \sigma^{v}_{\mathsf{abs}}(\dot{\delta})$	INT7SAb	$\upsilon^{p}_{\mathsf{abs}}(\int_{v \in V} F(v)) = \int_{v \in V} \upsilon^{p}_{\mathsf{abs}}(F(v))$	SATO6
$\sup V = p,\ p \notin V \Rightarrow$		$\overline{\upsilon}^{p}_{\mathsf{abs}}(\int_{v \in V} F(v)) = \int_{v \in V} \overline{\upsilon}^{p}_{\mathsf{abs}}(F(v))$	SAI6
$\int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\tilde{\delta}) = \sigma^{p}_{\mathsf{abs}}(\dot{\delta})$	INT8SAa	$\nu_{\mathsf{abs}}(\int_{v \in V} F(v)) = \int_{v \in V} \nu_{\mathsf{abs}}(F(v))$	SAU5
V, W unbounded $\Rightarrow$			
$\int_{v \in V} \sigma^{v}_{\mathsf{abs}}(\tilde{\delta}) = \int_{v \in W} \sigma^{v}_{\mathsf{abs}}(\dot{\delta})$	INT8SAb		

Table 17. Additional axioms for $\mathrm{ACP}^{\mathrm{sat}}\mathrm{I}\surd$ ($p \geq 0$)

$\surd_{\mathsf{s}} v \,.\, F(v) = \surd_{\mathsf{s}} w \,.\, F(w)$	SIA1	$\upsilon^{p}_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, \upsilon^{p}_{\mathsf{abs}}(F(v))$	SIA10
$\overline{\upsilon}^{p}_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \overline{\upsilon}^{p}_{\mathsf{abs}}(F(p))$	SIA2	$(\surd_{\mathsf{s}} v \,.\, F(v)) \mathbin{\|\!\_} x = \surd_{\mathsf{s}} v \,.\, (F(v) \mathbin{\|\!\_} \overline{\upsilon}^{v}_{\mathsf{abs}}(x))$	SIA11
$\surd_{\mathsf{s}} v \,.\, (\surd_{\mathsf{s}} w \,.\, K(v,w)) = \surd_{\mathsf{s}} v \,.\, K(v,v)$	SIA3	$x \mathbin{\|\!\_} (\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, (\overline{\upsilon}^{v}_{\mathsf{abs}}(x) \mathbin{\|\!\_} F(v))$	SIA12
$x = \surd_{\mathsf{s}} v \,.\, x$	SIA4	$(\surd_{\mathsf{s}} v \,.\, F(v)) \mid x = \surd_{\mathsf{s}} v \,.\, (F(v) \mid \overline{\upsilon}^{v}_{\mathsf{abs}}(x))$	SIA13
$(\forall v \in \mathbb{R}^{\geq} \bullet \overline{\upsilon}^{v}_{\mathsf{abs}}(x) = \overline{\upsilon}^{v}_{\mathsf{abs}}(y)) \Rightarrow x = y$	SIA5	$x \mid (\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, (\overline{\upsilon}^{v}_{\mathsf{abs}}(x) \mid F(v))$	SIA14
$\sigma^{p}_{\mathsf{abs}}(\tilde{a}) \cdot x = \sigma^{p}_{\mathsf{abs}}(\tilde{a}) \cdot \overline{\upsilon}^{p}_{\mathsf{abs}}(x)$	SIA6	$\partial_H(\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, \partial_H(F(v))$	SIA15
$\sigma^{p}_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \sigma^{p}_{\mathsf{abs}}(F(0))$	SIA7	$\nu_{\mathsf{abs}}(\surd_{\mathsf{s}} v \,.\, F(v)) = \surd_{\mathsf{s}} v \,.\, \nu_{\mathsf{abs}}(F(v))$	SIA16
$(\surd_{\mathsf{s}} v \,.\, F(v)) + x = \surd_{\mathsf{s}} v \,.\, (F(v) + \overline{\upsilon}^{v}_{\mathsf{abs}}(x))$	SIA8	$\int_{v \in V} (\surd_{\mathsf{s}} w \,.\, K(v,w)) =$	
$(\surd_{\mathsf{s}} v \,.\, F(v)) \cdot x = \surd_{\mathsf{s}} v \,.\, (F(v) \cdot x)$	SIA9	$\surd_{\mathsf{s}} w \,.\, (\int_{v \in V} K(v,w))$ if $v \neq w$	SIA17

References

[1] Baeten, J.C.M., Bergstra, J.A.: Real Time Process Algebra, *Formal Aspects of Computing*, **3**(2), 1991, 142–188.

[2] Baeten, J.C.M., Bergstra, J.A.: Discrete Time Process Algebra (extended abstract), *CONCUR'92* (W.R. Cleaveland, Ed.), LNCS 630, Springer-Verlag, Berlin, 1992, Full version: Report P9208b, Programming Research Group, University of Amsterdam.

[3] Baeten, J.C.M., Bergstra, J.A.: Real Space Process Algebra, *Formal Aspects of Computing*, **5**(6), 1993, 481–529.

[4] Baeten, J.C.M., Bergstra, J.A.: Real Time Process Algebra with Infinitesimals, *Algebra of Communicating Processes 1994* (A. Ponse, C. Verhoef, S.F.M. van Vlijmen, Eds.), Workshop in Computing Series, Springer-Verlag, Berlin, 1995.

[5] Baeten, J.C.M., Bergstra, J.A.: Discrete Time Process Algebra, *Formal Aspects of Computing*, **8**(2), 1996, 188–208.

[6] Baeten, J.C.M., Middelburg, C.A.: Process Algebra with Timing: Real Time and Discrete Time, in: *Handbook of Process Algebra* (J.A. Bergstra, A. Ponse, S.A. Smolka, Eds.), Elsevier, Amsterdam, 2001, 627–684.

[7] Baeten, J.C.M., Middelburg, C.A.: *Process Algebra with Timing*, EATCS Monographs Series, Springer Verlag, Berlin, 2002.

[8] Baeten, J.C.M., Verhoef, C.: Concrete Process Algebra, in: *Handbook of Logic in Computer Science, Volume IV* (S. Abramsky, D.M. Gabbay, T.S.E. Maibaum, Eds.), Oxford University Press, Oxford, 1995, 149–268.

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

[10] Chen, L.: An Interleaving Model for Real-Time Systems, *Symposium on Logical Foundations of Computer Science* (A. Nerode, M. Taitslin, Eds.), LNCS 620, Springer-Verlag, Berlin, 1992.

[11] Goldblatt, R.: *Lectures on the Hyperreals: An Introduction to Nonstandard Analysis*, Springer-Verlag, Berlin, 1998.

[12] Middelburg, C.A.: Variable Binding Operators in Transition System Specifications, *Journal of Logic and Algebraic Programming*, **47**(1), 2001, 15–45.

[13] Middelburg, C.A.: An Alternative Formulation of Operational Conservativity with Binding Terms, *Journal of Logic and Algebraic Programming*, 2003, To appear.

[14] Middelburg, C.A.: Revisiting Timing in Process Algebra, *Journal of Logic and Algebraic Programming*, **54**(1–2), 2003, 109–127.

[15] Moller, F., Tofts, C.: A Temporal Calculus of Communicating Systems, *CONCUR'90* (J.C.M. Baeten, J.W. Klop, Eds.), LNCS 458, Springer-Verlag, Berlin, 1990.

[16] Nicollin, X., Sifakis, J.: The Algebra of Timed Processes ATP: Theory and Application, *Information and Computation*, **114**(1), 1994, 131–178.

[17] Robinson, A.: *Non-standard Analysis*, Revised edition, Princeton University Press, Princeton, 1996.

[18] Wang Yi: Real-Time Behaviour of Asynchronous Agents, *CONCUR'90* (J.C.M. Baeten, J.W. Klop, Eds.), LNCS 458, Springer-Verlag, Berlin, 1990.