

Real time process algebra with time-dependent conditions

Citation for published version (APA):

Baeten, J. C. M., & Middelburg, C. A. (2000). *Real time process algebra with time-dependent conditions*. (Logic Group Preprint Series; Vol. 200). Utrecht University.

Document status and date:

Published: 01/01/2000

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Real Time Process Algebra with Time-dependent Conditions

J.C.M. Baeten¹ and C.A. Middelburg^{1,2}

¹ Computing Science Department, Eindhoven University of Technology
Eindhoven, the Netherlands

² Department of Philosophy, Utrecht University
Utrecht, the Netherlands

{josb,keesm}@win.tue.nl

Abstract

We extend the main real time version of ACP presented in [6] with conditionals in which the condition depends on time. This extension facilitates flexible dependence of process behaviour on initialization time. We show that the conditions concerned generalize the conditions introduced earlier in a discrete time setting in [4].

Keywords & Phrases: process algebra, ACP, real time, discrete time, absolute timing, relative timing, parametric timing, initialization, time-dependent conditions, conditionals.

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

1 Introduction

Algebraic concurrency theories such as ACP [9, 8, 7], CCS [16, 17] and CSP [10, 14] have been extended to deal with time-dependent behaviour in various ways. In [6], we presented results of a systematic study of some of the most important issues relevant to dealing with time-dependent behaviour of processes – viz. absolute vs relative timing, continuous vs discrete time scale, and separation vs combination of execution of actions and passage of time – in the setting of ACP. We presented real time and discrete time versions of ACP with both absolute timing and relative timing, starting with a new real time version of ACP with absolute timing called ACP^{sat} . We demonstrated that ACP^{sat} extended with integration and initial abstraction generalizes the presented real time version with relative timing and the presented discrete time version with absolute timing. Integration provides for alternative composition over a continuum of alternatives; and initial abstraction, being λ -calculus-like functional abstraction where the parameter is initialization time, provides for simple parametric timing. We focussed on versions of ACP with timing where execution of actions and passage of time are separated, but explained how versions with time stamping of actions can be obtained.

The real time versions of ACP presented in [6], unlike those presented in [2] and [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 [19] and the different versions of CCS with timing [11, 18, 21]. 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. In [2] and [3], ways to deal with independent actions are proposed where such actions take place at the same point in time by treating it as a special case of communication. This is, however, a real burden in the description and the analysis of the systems concerned.

In this paper we extend ACP^{sat} further with conditionals in which the condition depends on time. The conditions concerned generalize the conditions introduced earlier in [4] to extend discrete time versions of ACP with conditionals in which the condition depends on time. The extension allows an interesting expansion property of processes with parametric timing, called time spectrum expansion, to be expressed. It is practically useful as well, because it facilitates flexible dependence of process behaviour on initialization time. We also extend the discrete time counterpart of ACP^{sat} presented in [6] with conditionals in which the condition depends on time. In this case, the conditions are essentially the same as the conditions introduced earlier in [4].

In [6], our aim was to present a coherent collection of algebraic concurrency theories generalizing ACP that deal with time-dependent behaviour in different ways. In this paper, we extend the main real time and discrete time versions of ACP presented in [6] with conditionals in which the condition depends on time. By showing that the discrete time version with conditionals can be embedded in the real time version with conditionals, we demonstrate that the extensions with conditionals do not destroy the coherence.

The structure of this paper is as follows. First, we review ACP^{sat} and its extension with integration and initial abstraction in Sections 2. Then, in Section 3, we add conditionals in which the condition depends on time to this real time version of ACP. After that, in Section 4, we first briefly review the discrete time counterpart of ACP^{sat} and then add conditionals in which the condition depends on time to this discrete time version of ACP. In Section 5, we show that the discrete time version with conditionals can be embedded in the real time version with conditionals.

2 Real time process algebra with absolute timing

In this section, we review ACP^{sat} , the real time process algebra with absolute timing introduced in [6], and its extension with integration and initial abstraction. A detailed account of this real time version of ACP and these extensions is given in [6]. The axioms and operational semantics rules – extracted from [6] – are given in Appendix A.

In case of ACP^{sat} , it is assumed that a theory of the non-negative real numbers has been given. Its signature has to include the constant $0 : \rightarrow \mathbb{R}_{\geq 0}$, the operator $+$: $\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$, and the predicates \leq : $\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ and $=$: $\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$. In addition, this theory has to include axioms that characterize $+$ as a commutative and associative operation with 0 as a neutral element and \leq as a total ordering that has

0 as its least element and that is preserved by $+$.

In ACP^{sat} , as in the other versions of ACP with timing presented in this paper, it is assumed that a fixed but arbitrary set \mathbf{A} of *actions* has been given. It is also assumed that a fixed but arbitrary *communication function*, i.e. a partial, commutative and associative function $\gamma : \mathbf{A} \times \mathbf{A} \rightarrow \mathbf{A}$, has been given. The function γ is regarded to give the result of the synchronous execution of any two actions for which this is possible, and to be undefined otherwise. The weak restrictions on γ allow many kinds of communication between parallel processes to be modeled.

First, in Section 2.1, we treat BPA^{sat} , basic standard real time process algebra with absolute timing, in which parallelism and communication are not considered. After that, in Section 2.2, BPA^{sat} is extended to ACP^{sat} to deal with parallelism and communication as well. In Section 2.3, integration and initial abstraction are added to ACP^{sat} . Finally, some useful additional axioms, derivable for closed terms, and elimination results are given in Section 2.4.

2.1 Basic process algebra

In BPA^{sat} , we have the sort \mathbf{P} of processes, the *urgent action* constants $\tilde{a} : \rightarrow \mathbf{P}$ (one for each $a \in \mathbf{A}$), the *urgent deadlock* constant $\tilde{\delta} : \rightarrow \mathbf{P}$, the *immediate deadlock* constant $\dot{\delta} : \rightarrow \mathbf{P}$, the *alternative composition* operator $+$: $\mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$, the *sequential composition* operator \cdot : $\mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$, the *absolute delay* operator $\sigma_{\text{abs}} : \mathbb{R}_{\geq 0} \times \mathbf{P} \rightarrow \mathbf{P}$, the *absolute time-out* operator $v_{\text{abs}} : \mathbb{R}_{\geq 0} \times \mathbf{P} \rightarrow \mathbf{P}$, and the *absolute initialization* operator $\bar{v}_{\text{abs}} : \mathbb{R}_{\geq 0} \times \mathbf{P} \rightarrow \mathbf{P}$.

The process \tilde{a} is only capable of performing action a , immediately followed by successful termination, at time 0. The process $\tilde{\delta}$, although existing at time 0, is incapable of doing anything. The process $\dot{\delta}$ stands for a process that exhibits inconsistent timing at time 0. This means that $\dot{\delta}$, different from $\tilde{\delta}$, does not exist at time 0 and hence causes a time stop at time 0. The process $\sigma_{\text{abs}}^p(x)$ is the process x shifted in time by p . Thus, the process $\sigma_{\text{abs}}^p(\tilde{\delta})$ is capable of idling from time 0 upto and including time p – and at time p it gets incapable of doing anything – whereas the process $\sigma_{\text{abs}}^p(\dot{\delta})$ is only capable of idling from time 0 upto, but not including, time p . So $\sigma_{\text{abs}}^p(\dot{\delta})$ can not reach time p . The process $x \cdot y$ is the process x followed upon successful termination by the process y . The process $x + y$ is the process that proceeds with either the process x or the process y , but not both. As in the untimed case, the choice is resolved upon execution of the first action, and not before. We also have two auxiliary operators: v_{abs} and \bar{v}_{abs} . The process $v_{\text{abs}}^p(x)$ is the part of x that starts to perform actions before time p . The process $\bar{v}_{\text{abs}}^p(x)$ is the part of x that starts to perform actions at time p or later.

We assume that an infinite set of variables of sort \mathbf{P} has been given. Given the signature of BPA^{sat} , terms of BPA^{sat} are constructed in the usual way. We will in general use infix notation for binary operators. The need to use parentheses is further reduced by ranking the precedence of the binary operators. Throughout this paper we adhere to the following precedence rules: (i) the operator \cdot has the highest precedence amongst the binary operators, (ii) the operator $+$ has the lowest precedence amongst the binary operators, and (iii) all other binary operators have the same precedence. We will also use the following abbreviation. Let $(t_i)_{i \in \mathcal{I}}$ be an indexed set of terms of BPA^{sat} where $\mathcal{I} = \{i_1, \dots, i_n\}$. Then we write $\sum_{i \in \mathcal{I}} t_i$ for $t_{i_1} + \dots + t_{i_n}$. We further

use the convention that $\sum_{i \in \mathcal{I}} t_i$ stands for $\dot{\delta}$ if $\mathcal{I} = \emptyset$.

We denote variables by x, x', y, y', \dots . We use a, a', b, b', \dots to denote elements of $\mathbf{AU}\{\delta\}$ in the context of an equation, and elements of \mathbf{A} in the context of an operational semantics rule. Furthermore, we use H to denote a subset of \mathbf{A} . We denote elements of $\mathbb{R}_{\geq 0}$ by p, p', q, q' and elements of $\mathbb{R}_{> 0}$ by r, r' . We write \mathbf{A}_δ for $\mathbf{A} \cup \{\delta\}$.

Axiom system The axiom system of BPA^{sat} consists of the equations given in Table 15. For a discussion of the axioms of BPA^{sat} , see [6].

The following lemmas from [6] are useful in proofs. They were, for example, used there to shorten the calculations in the proof of an embedding theorem.

Lemma 1 *In BPA^{sat} and ACP^{sat} , as well as in the further extensions with restricted integration and initial abstraction:*

1. *the equation $t = v_{\text{abs}}^p(t) + \bar{v}_{\text{abs}}^p(t)$ is derivable for all closed terms t such that $t = \bar{v}_{\text{abs}}^0(t)$ and $t = t + \sigma_{\text{abs}}^p(\dot{\delta})$;*
2. *the equations $t = v_{\text{abs}}^p(t)$ and $\bar{v}_{\text{abs}}^p(t) = \sigma_{\text{abs}}^p(\dot{\delta})$ are derivable for all closed terms t such that $t = \bar{v}_{\text{abs}}^0(t)$ and $t \neq t + \sigma_{\text{abs}}^p(\dot{\delta})$.*

Lemma 2 *In BPA^{sat} and ACP^{sat} , as well as in the further extensions with restricted integration and initial abstraction, for each $p \in \mathbb{R}_{\geq 0}$ and each closed term t , there exists a closed term t' such that $\bar{v}_{\text{abs}}^p(t) = \sigma_{\text{abs}}^p(t')$ and $t' = \bar{v}_{\text{abs}}^0(t')$.*

Lemma 1 indicates that a process that is able to reach time p can be regarded as being the alternative composition of the part that starts to perform actions before p and the part that starts to perform actions at p or later. Lemma 2 shows that the part of a process that starts to perform actions at time p or later can always be regarded as a process shifted in time by p .

Semantics A *real time transition system* over \mathbf{A} consists of a set of *states* \mathbf{S} , a *root state* $\rho \in \mathbf{S}$ and four kinds of relations on states:

- a binary relation $\langle -, p \rangle \xrightarrow{a} \langle -, p \rangle$ for each $a \in \mathbf{A}$, $p \in \mathbb{R}_{\geq 0}$,
- a unary relation $\langle -, p \rangle \xrightarrow{a} \langle \surd, p \rangle$ for each $a \in \mathbf{A}$, $p \in \mathbb{R}_{\geq 0}$,
- a binary relation $\langle -, p \rangle \xrightarrow{r} \langle -, q \rangle$ for each $r \in \mathbb{R}_{> 0}$, $p, q \in \mathbb{R}_{\geq 0}$ where $q = p + r$,
- a unary relation $\text{ID}(-, p)$ for each $p \in \mathbb{R}_{\geq 0}$;

satisfying

1. if $\langle s, p \rangle \xrightarrow{r+r'} \langle s', q \rangle$, $r, r' > 0$, then there is a s'' such that $\langle s, p \rangle \xrightarrow{r} \langle s'', p+r \rangle$ and $\langle s'', p+r \rangle \xrightarrow{r'} \langle s', q \rangle$;
2. if $\langle s, p \rangle \xrightarrow{r} \langle s'', p+r \rangle$ and $\langle s'', p+r \rangle \xrightarrow{r'} \langle s', q \rangle$, then $\langle s, p \rangle \xrightarrow{r+r'} \langle s', q \rangle$.

The four kinds of relations are called *action step*, *action termination*, *time step* and *immediate deadlock* relations, respectively. We write $\text{RTTS}(\mathbf{A})$ for the set of all real time transition systems over \mathbf{A} .

We shall associate a transition system in $\text{RTTS}(\mathbf{A})$ with a closed term t of BPA^{sat} by taking the set of closed terms of BPA^{sat} as set of states, the closed term t as root state, and the action step, action termination, time step and immediate deadlock

relations defined in Table 20 using rules in the style of Plotkin [20]. A semantics given in this way is called a structural operational semantics. On the basis of the operational semantics of BPA^{sat} , the operators of BPA^{sat} can also be directly defined on the set of real time transition systems in a straightforward way. Note that, by taking closed terms as states, the relations can be explained as follows:

- $\langle t, p \rangle \xrightarrow{a} \langle t', p \rangle$: process t is capable of first performing action a at time p and then proceeding as process t' ;
- $\langle t, p \rangle \xrightarrow{a} \langle \surd, p \rangle$: process t is capable of first performing action a at time p and then terminating successfully;
- $\langle t, p \rangle \xrightarrow{r} \langle t', q \rangle$: process t is capable of first idling from time p to time q and then proceeding as process t' ;
- $\text{ID}(t, p)$: process t is not capable of reaching time p .

The rules for the operational semantics have the form $\frac{h_1, \dots, h_m, s}{c_1, \dots, c_n}$, where s is optional. They are to be read as “if h_1 and \dots and h_m then c_1 and \dots and c_n , provided s ”. The conclusions c_1, \dots, c_n are positive formulas of the form $\langle t, p \rangle \xrightarrow{a} \langle t', p \rangle$, $\langle t, p \rangle \xrightarrow{a} \langle \surd, p \rangle$, $\langle t, p \rangle \xrightarrow{r} \langle t', q \rangle$ or $\text{ID}(t, p)$, where t and t' are open terms of BPA^{sat} . The premises h_1, \dots, h_m are positive formulas of the above forms or negative formulas of the form $\neg \text{ID}(t, p)$. The rules are actually rule schemas. The optional s is a side-condition restricting the actions over which a, b and c range and the non-negative real numbers over which p, q and r range.

By identifying bisimilar processes we obtain our preferred model of BPA^{sat} . One process is (strongly) bisimilar to another process means that if one of the processes is capable of doing a certain step, i.e. performing a certain action at a certain time or idling from a certain time to another, and next going on as a certain subsequent process then the other process is capable of doing the same step and next going on as a process bisimilar to the subsequent process. More precisely, a *bisimulation* on $\text{RTTS}(\mathbf{A})$ is a symmetric binary relation R on the set of states \mathbf{S} such that:

1. if $R(s, t)$ and $\langle s, p \rangle \xrightarrow{a} \langle s', p \rangle$, then there is a t' such that $\langle t, p \rangle \xrightarrow{a} \langle t', p \rangle$ and $R(s', t')$;
2. if $R(s, t)$, then $\langle s, p \rangle \xrightarrow{a} \langle \surd, p \rangle$ iff $\langle t, p \rangle \xrightarrow{a} \langle \surd, p \rangle$;
3. if $R(s, t)$ and $\langle s, p \rangle \xrightarrow{r} \langle s', q \rangle$, then there is a t' such that $\langle t, p \rangle \xrightarrow{r} \langle t', q \rangle$ and $R(s', t')$;
4. if $R(s, t)$, then $\text{ID}(s, p)$ iff $\text{ID}(t, p)$.

We say that two closed terms s and t are *bisimilar*, written $s \Leftrightarrow t$, if there exists a bisimulation R such that $R(s, t)$. Bisimulation equivalence is a congruence for the operators of BPA^{sat} . For this reason, the operators of BPA^{sat} can be defined on the set of bisimulation equivalence classes. We can prove that this results in a model for BPA^{sat} , i.e. all equations derivable in BPA^{sat} hold. In other words, the axioms of BPA^{sat} form a sound axiomatization for the model based on bisimulation equivalence classes. As in the case of the other axiomatizations presented in this paper, we leave it as an open problem whether the axioms of BPA^{sat} form a complete axiomatization for this model.

2.2 Algebra of communicating processes

In ACP^{sat} , we have, in addition to the constants and operators of BPA^{sat} , the *parallel composition* operator $\parallel: \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$, the *left merge* operator $\ll: \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$, the *communication merge* operator $|: \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$, the *encapsulation* operators $\partial_H: \mathbf{P} \rightarrow \mathbf{P}$ (for each $H \subseteq \mathbf{A}$), and the *absolute urgent initialization* operator $\nu_{\text{abs}}: \mathbf{P} \rightarrow \mathbf{P}$.

The process $x \parallel y$ is the process that proceeds with the processes x and y in parallel. It may start to perform actions by (i) performing an action of x if x can do so before or at the ultimate time for y to start performing actions or to deadlock, (ii) performing an action of y if y can do so before or at the ultimate time for x to start performing actions or to deadlock or (iii) performing an action of x and an action of y synchronously if x and y can do so at the same time. Furthermore, we have the encapsulation operators ∂_H (one for each $H \subseteq \mathbf{A}$) which turns all urgent actions \tilde{a} , where $a \in H$, into $\tilde{\delta}$. As in ACP, we also have the auxiliary operators \ll and $|$ to get a finite axiomatization of the parallel composition operator. The processes $x \parallel y$ and $x \ll y$ are the same except that $x \ll y$ must start to perform actions by performing an action of x . The processes $x | y$ and $x \parallel y$ are the same except that $x | y$ must start to perform actions by performing an action of x and an action of y synchronously. In case of ACP^{sat} , one additional auxiliary operator is needed: ν_{abs} . The process $\nu_{\text{abs}}(x)$ is the part of process x that starts to perform actions at time 0.

Axiom system The axiom system of ACP^{sat} consists of the axioms of BPA^{sat} and the equations given in Table 16. For a discussion of the axioms of ACP^{sat} , see [6].

Semantics The structural operational semantics of ACP^{sat} is described by the rules for BPA^{sat} and the rules given in Table 21. For a discussion of some of these rules, see [6]. Bisimulation equivalence is also a congruence for the additional operators of ACP^{sat} . Therefore, these operators can be defined on the set of bisimulation equivalence classes as well. As in the case of BPA^{sat} , we can prove that this results in a model for ACP^{sat} .

2.3 Integration and initial abstraction

In this subsection, we review the extension of ACP^{sat} with integration and initial abstraction. The extension with integration is needed to be able to embed discrete time process algebras. The extension with initial abstraction is needed to be able to embed process algebras with relative timing.

Integration and initial abstraction are both variable binding operators. Following e.g. [12], we will introduce *variable binding operators* by a declaration of the form $f: S_{11}, \dots, S_{1k_1} \cdot S_1 \times \dots \times S_{n1}, \dots, S_{nk_n} \cdot S_n \rightarrow S$. Hereby is indicated that f combines an operator $f^*: ((S_{11} \times \dots \times S_{1k_1}) \rightarrow S_1) \times \dots \times ((S_{n1} \times \dots \times S_{nk_n}) \rightarrow S_n) \rightarrow S$ with λ -calculus-like functional abstraction, binding k_i variables ranging over S_{i1}, \dots, S_{ik_i} in the i th argument ($0 \leq i \leq n$). Applications of f have the following form: $f(x_{11}, \dots, x_{1k_1} \cdot t_1, \dots, x_{n1}, \dots, x_{nk_n} \cdot t_n)$, where each x_{ij} is a variable of sort S_{ij} and each t_i is a term of sort S_i .

Integration requires a more extensive theory of the non-negative real numbers than the minimal theory sketched at the beginning of Section 2 (page 2). In the first place, it has to include a theory of sets of non-negative real numbers that makes it possible to

deal with set membership and set equality. Besides, the theory should cover suprema of sets of non-negative real numbers.

First, ACP^{sat} is extended with integration. After that, initial abstraction is added.

Integration

In $\text{ACP}^{\text{sat}}\text{I}$, we have, in addition to the constants and operators of ACP^{sat} , the *integration* (variable-binding) operator $\int : \mathcal{P}(\mathbb{R}_{\geq 0}) \times \mathbb{R}_{\geq 0} . \mathbf{P} \rightarrow \mathbf{P}$. The integration operator \int provides for alternative composition over a continuum of alternatives. That is, $\int_{v \in V} P$, where v is a variable ranging over $\mathbb{R}_{\geq 0}$, $V \subseteq \mathbb{R}_{\geq 0}$ and P is a term that may contain free variables, proceeds as one of the alternatives $P[p/v]$ for $p \in V$. Obviously, we could first have added integration to BPA^{sat} , resulting in $\text{BPA}^{\text{sat}}\text{I}$, and then have extended $\text{BPA}^{\text{sat}}\text{I}$ to deal with parallelism and communication.

We assume that an infinite set of *time variables* ranging over $\mathbb{R}_{\geq 0}$ has been given, and denote them by v, w, \dots . Furthermore, we use V, W, \dots to denote subsets of $\mathbb{R}_{\geq 0}$. We denote terms of $\text{ACP}^{\text{sat}}\text{I}$ by P, Q, \dots . We will use the following notational convention. We write $\int_{v \in V} P$ for $\int(V, v . P)$.

Axiom system The axiom system of $\text{ACP}^{\text{sat}}\text{I}$ consists of the axioms of ACP^{sat} and the equations given in Table 17. Axioms INT1-INT6 are the crucial axioms of integration. They reflect the informal explanation given above.

Semantics The structural operational semantics of $\text{ACP}^{\text{sat}}\text{I}$ is described by the rules for ACP^{sat} and the rules given in Table 22. Bisimulation equivalence is also a congruence for the integration operator. Hence, this operator can be defined on the set of bisimulation equivalence classes as well. As in the case of BPA^{sat} and ACP^{sat} , we can prove that this results in a model for $\text{ACP}^{\text{sat}}\text{I}$. We will call this model \mathbf{M}_A .

Initial abstraction

In $\text{ACP}^{\text{sat}}\text{I}\checkmark$, we have, in addition to the constants and operators of $\text{ACP}^{\text{sat}}\text{I}$, the *initial abstraction* (variable-binding) operator $\sqrt{s} : \mathbb{R}_{\geq 0} . \mathbf{P}^* \rightarrow \mathbf{P}^*$. The initial abstraction operator \sqrt{s} provides for simple parametric timing: $\sqrt{s} v . F$, where v is a variable ranging over $\mathbb{R}_{\geq 0}$ and F is a term that may contain free variables, proceeds as $F[p/v]$ if initialized at time $p \in \mathbb{R}_{\geq 0}$. This means that $\sqrt{s} v . F$ denotes a function $f : \mathbb{R}_{\geq 0} \rightarrow \mathbf{P}$ that satisfies $f(p) = \bar{v}^p_{\text{abs}}(f(p))$ for all $p \in \mathbb{R}_{\geq 0}$. In $\text{ACP}^{\text{sat}}\text{I}\checkmark$, i.e. $\text{ACP}^{\text{sat}}\text{I}$ with initial abstraction, the sort \mathbf{P} of processes is replaced by the sort \mathbf{P}^* of parametric time processes. Of course, it is also possible to add the initial abstraction operator to ACP^{sat} , resulting in a theory $\text{ACP}^{\text{sat}}\checkmark$.

We now use x, y, \dots to denote variables of sort \mathbf{P}^* . Terms of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ are denoted by F, G, \dots . We will use the following notational convention. We write $\sqrt{s} v . F$ for $\sqrt{s}(v . F)$.

Axiom system The axiom system of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ consists of the axioms of $\text{ACP}^{\text{sat}}\text{I}$ and the equations given in Table 18. Axioms SIA1-SIA6 are the crucial axioms of initial abstraction. Axioms SIA1 and SIA2 are similar to the α - and β -conversion rules of λ -calculus. Axiom SIA3 points out that multiple initial abstractions can simply be replaced by one. Axiom SIA4 shows that processes with absolute timing can be treated as special cases of processes with parametric timing: they do not vary with

different initialization times. Axiom SIA5 is an extensionality axiom. Axiom SIA6 expresses that in case a process performs an action and then proceeds as another process, the initialization time of the latter process is the time at which the action is performed.

Semantics On the basis of the rules for its operational semantics, the operators of $\text{ACP}^{\text{sat}}\text{I}$ can also be directly defined on real time transition systems in a straightforward way. We will now describe a model of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ in terms of these operators.

We have to extend $\text{RTTS}(\mathbf{A})$ to the function space

$$\text{RTTS}^*(\mathbf{A}) = \{f : \mathbb{R}_{\geq 0} \rightarrow \text{RTTS}(\mathbf{A}) \mid \forall p \in \mathbb{R}_{\geq 0} \bullet f(p) = \bar{v}_{\text{abs}}^p(f(p))\}$$

of *real time transition systems with parametric timing*. In Table 23, the constants and operators of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ are defined on $\text{RTTS}^*(\mathbf{A})$. We say that $f, g \in \text{RTTS}^*(\mathbf{A})$ are bisimilar if for all $p \in \mathbb{R}_{\geq 0}$, there exists a bisimulation R such that $R(f(p), g(p))$. We obtain a model of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ by defining all operators on the set of bisimulation equivalence classes. We will call this model $\mathbf{M}_{\mathbf{A}}^*$. Notice that $f \in \text{RTTS}^*(\mathbf{A})$ corresponds to a process that can be written with only the constants and operators of $\text{ACP}^{\text{sat}}\text{I}$ iff $\bar{v}_{\text{abs}}^0(f) = f$. In fact, $\mathbf{M}_{\mathbf{A}}$ is isomorphic to a subalgebra of $\mathbf{M}_{\mathbf{A}}^*$.

2.4 Miscellaneous

Standard initialization axioms

In Table 19, some equations concerning initialization and time-out are given that hold in the model $\mathbf{M}_{\mathbf{A}}^*$, and that are derivable for closed terms of $\text{ACP}^{\text{sat}}\text{I}\checkmark$. We will use these axioms in proofs in subsequent sections.

Using the standard initialization axioms, the following can easily be derived for all terms F and F' :

$$(\sqrt{s}v . F) \square (\sqrt{s}v . F') = \sqrt{s}v . (F \square F') \text{ DISTR}\square$$

for $\square = +, \parallel, \llbracket, \mid$. In other words, initial abstraction distributes over $+$, \parallel , \llbracket and \mid . This fact is a useful aid to shorten the calculations needed in proofs.

Elimination results

We can prove that the auxiliary operators v_{abs} and \bar{v}_{abs} , as well as sequential compositions in which the form of the first operand is not \tilde{a} ($a \in \mathbf{A}$) and alternative compositions in which the form of the first operand is $\sigma_{\text{abs}}^p(t)$, can be eliminated in closed terms of $\text{BPA}^{\text{sat}}\text{I}$ with a restricted form of integration. Basically, this restriction means that in terms of the form $\int_{v \in V} P$, V is an interval of which the bounds are given by linear expressions over time variables and P is of the form $\sigma_{\text{abs}}^v(\tilde{a})$ or $\sigma_{\text{abs}}^v(\tilde{a}) \cdot t$ ($a \in \mathbf{A}_\delta$). This restricted form of integration is essentially the same as prefix integration from [15]. The terms that remain after exhaustive elimination are called the basic terms over BPA^{sat} with restricted integration. We can also prove that the operators $\parallel, \llbracket, \mid, \partial_H$ and ν_{abs} can be eliminated in closed terms of ACP^{sat} with restricted integration. Because of these elimination results, we are permitted to use induction on the structure of basic terms over BPA^{sat} with restricted integration to prove statements for all closed terms of ACP^{sat} with restricted integration.

The elimination results for $\text{ACP}^{\text{sat}}\checkmark$ with restricted integration are essentially the same as the ones for ACP^{sat} with restricted integration. Besides, all closed terms of $\text{ACP}^{\text{sat}}\checkmark$ with restricted integration can be written in the form $\sqrt{s}v \cdot F$ where F is a basic term over BPA^{sat} with restricted integration.

Examples We give some examples of a closed term of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ with restricted integration, the corresponding term of the form $\sqrt{s}v \cdot F$ where F is a basic term and, if possible, the corresponding basic term without initial abstraction:

$$\begin{aligned} \sqrt{s}v \cdot v_{\text{abs}}^{v+2.3}(\sqrt{s}w \cdot \sigma_{\text{abs}}^w(\tilde{a})) &= \sqrt{s}v \cdot \sigma_{\text{abs}}^v(\tilde{a}) \\ \sqrt{s}v \cdot \bar{v}_{\text{abs}}^{v+2.3}(\sqrt{s}w \cdot \sigma_{\text{abs}}^w(\tilde{a})) &= \sqrt{s}v \cdot \sigma_{\text{abs}}^{v+2.3}(\tilde{a}) \\ \bar{v}_{\text{abs}}^{3.9}(\sqrt{s}v \cdot \bar{v}_{\text{abs}}^{v+2.3}(\int_{w \in [6,6.1]} \sigma_{\text{abs}}^w(\tilde{a}))) &= \sqrt{s}v \cdot \sigma_{\text{abs}}^{6.2}(\delta) = \sigma_{\text{abs}}^{6.2}(\delta) \\ \bar{v}_{\text{abs}}^{3.6}(\sqrt{s}v \cdot \bar{v}_{\text{abs}}^{v+2.3}(\int_{w \in [6,6.1]} \sigma_{\text{abs}}^w(\tilde{a}))) &= \sqrt{s}v \cdot \int_{w \in [6,6.1]} \sigma_{\text{abs}}^w(\tilde{a}) = \int_{w \in [6,6.1]} \sigma_{\text{abs}}^w(\tilde{a}) \end{aligned}$$

3 Conditionals with time-dependent conditions

In this section, we add a conditional operator with time-dependent conditions to $\text{ACP}^{\text{sat}}\text{I}\checkmark$. This operator facilitates flexible dependence of process behaviour on initialization time. The time-dependent conditions introduced here generalize the time-dependent conditions introduced in a discrete time setting in [4]. First, in Section 3.1, $\text{ACP}^{\text{sat}}\text{I}\checkmark$ is extended with time-dependent conditions and conditionals. After that, in Section 3.2, we describe a similar extension of $\text{ACP}^{\text{sat}}\text{I}$ and explain how it is related to the extension of $\text{ACP}^{\text{sat}}\text{I}\checkmark$. In Section 3.4, we give an example of the use of conditionals with time-dependent conditions. In Section 3.3, we describe the addition of recursion in outline to make understanding of the specifications given in Section 3.4 easier.

3.1 Parametric timing

We first introduce time-dependent conditions. We have the sort \mathbb{B}^* of time-dependent conditions, the *at time point* operator $\text{pt} : \mathbb{R} \rightarrow \mathbb{B}^*$, the *at time point greater than* operator $\text{pt}_{>} : \mathbb{R} \rightarrow \mathbb{B}^*$ (for technical reasons, it is convenient to use \mathbb{R} instead of $\mathbb{R}_{\geq 0}$ as the domain of these functions), the *logical* constants and operators $\mathbf{t} : \rightarrow \mathbb{B}^*$, $\mathbf{f} : \rightarrow \mathbb{B}^*$, $\neg : \mathbb{B}^* \rightarrow \mathbb{B}^*$, $\vee : \mathbb{B}^* \times \mathbb{B}^* \rightarrow \mathbb{B}^*$ and $\wedge : \mathbb{B}^* \times \mathbb{B}^* \rightarrow \mathbb{B}^*$, the *initialization* operator $\bar{v}_{\text{abs}} : \mathbb{R}_{\geq 0} \times \mathbb{B}^* \rightarrow \mathbb{B}^*$, and the *initial abstraction* operator $\sqrt{s} : \mathbb{R}_{\geq 0} \cdot \mathbb{B}^* \rightarrow \mathbb{B}^*$.

For a time-dependent condition b , $\bar{v}_{\text{abs}}^p(b)$ is either \mathbf{t} or \mathbf{f} , determined by whether b holds at time point p or not. For $p \in \mathbb{R}_{\geq 0}$, the condition $\text{pt}(p)$ holds only at time point p and the condition $\text{pt}_{>}(p)$ holds at all time points greater than p . For $r \in \mathbb{R}_{> 0}$, the condition $\text{pt}(-r)$ never holds and the condition $\text{pt}_{>}(-r)$ always holds – recall that all time points are in $\mathbb{R}_{\geq 0}$. The logical operators \neg , \vee and \wedge are defined on \mathbb{B}^* pointwise. Initial abstraction for conditions is similar to initial abstraction for processes.

We join time-dependent conditions with parametric time processes by means of the conditional operator $::\rightarrow$. In $\text{ACP}^{\text{sat}}\text{I}\checkmark\text{C}$, we have, in addition to the above-mentioned constants and operators on \mathbb{B}^* , the constants and operators of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ and the *conditional* operator $::\rightarrow : \mathbb{B}^* \times \mathbb{P}^* \rightarrow \mathbb{P}^*$.

Initialized at times where the condition b holds, the process $b ::\rightarrow x$ proceeds as the process x ; initialized at other times, $b ::\rightarrow x$ is the same as immediate deadlock.

We write b, b', \dots to denote variables of sort \mathbb{B}^* . Terms of sort \mathbb{B}^* are denoted by C, D, \dots . We will use the following abbreviations. We write $\mathbf{pt}_{\geq}(p)$ for $\mathbf{pt}_{>}(p) \vee \mathbf{pt}(p)$, $\mathbf{pt}_{\leq}(p)$ for $\neg \mathbf{pt}_{>}(p)$ and $\mathbf{pt}_{<}(p)$ for $\neg \mathbf{pt}_{\geq}(p)$. We further write $\sqrt{s}v . C$ for $\sqrt{s}(v . C)$.

Axiom system The axiom system of $\text{ACP}^{\text{sat}}\text{IVC}$ consists of the axioms of $\text{ACP}^{\text{sat}}\text{IV}$ and the equations given in Tables 1, 2 and 3. Axioms CSAI1-CSAI10 (Table 2)

| | | | |
|-------------------|-------|---|-------|
| $\neg t = f$ | BOOL1 | $t \vee b = t$ | BOOL4 |
| $\neg f = t$ | BOOL2 | $f \vee b = b$ | BOOL5 |
| $\neg \neg b = b$ | BOOL3 | $b \vee b' = b' \vee b$ | BOOL6 |
| | | $b \wedge b' = \neg(\neg b \vee \neg b')$ | BOOL7 |

Table 1: Axioms for logical operators

| | | | |
|---|--------|---|-------|
| $\bar{v}_{\text{abs}}^p(t) = t$ | CSAI1 | $\sqrt{s}w . D = \sqrt{s}v . D[v/w]$ | CSIA1 |
| $\bar{v}_{\text{abs}}^p(f) = f$ | CSAI2 | $\bar{v}_{\text{abs}}^p(\sqrt{s}v . C) = \bar{v}_{\text{abs}}^p(C[p/v])$ | CSIA2 |
| $\bar{v}_{\text{abs}}^p(\mathbf{pt}(p)) = t$ | CSAI3 | $\sqrt{s}v . (\sqrt{s}w . C) = \sqrt{s}v . C[v/w]$ | CSIA3 |
| $\bar{v}_{\text{abs}}^p(\mathbf{pt}(p - r)) = f$ | CSAI4 | $D = \sqrt{s}v . D$ | CSIA4 |
| $\bar{v}_{\text{abs}}^p(\mathbf{pt}(p + r)) = f$ | CSAI5 | $(\forall p \in \mathbb{R}_{\geq 0} \bullet \bar{v}_{\text{abs}}^p(b) = \bar{v}_{\text{abs}}^p(b')) \Rightarrow b = b'$ | CSIA5 |
| $\bar{v}_{\text{abs}}^p(\mathbf{pt}_{>}(p - r)) = t$ | CSAI6 | | |
| $\bar{v}_{\text{abs}}^p(\mathbf{pt}_{>}(p + q)) = f$ | CSAI7 | $\neg(\sqrt{s}v . C) = \sqrt{s}v . \neg C$ | CSIA6 |
| $\bar{v}_{\text{abs}}^p(\neg b) = \neg \bar{v}_{\text{abs}}^p(b)$ | CSAI8 | $(\sqrt{s}v . C) \wedge D = \sqrt{s}v . (C \wedge \bar{v}_{\text{abs}}^v(D))$ | CSIA7 |
| $\bar{v}_{\text{abs}}^p(b \wedge b') = \bar{v}_{\text{abs}}^p(b) \wedge \bar{v}_{\text{abs}}^p(b')$ | CSAI9 | $(\sqrt{s}v . C) \vee D = \sqrt{s}v . (C \vee \bar{v}_{\text{abs}}^v(D))$ | CSIA8 |
| $\bar{v}_{\text{abs}}^p(b \vee b') = \bar{v}_{\text{abs}}^p(b) \vee \bar{v}_{\text{abs}}^p(b')$ | CSAI10 | | |

Table 2: Axioms for conditions ($p, q \geq 0, r > 0, v$ not free in D)

reflect the intended meaning of the initialization operator on conditions, viz. evaluation at initialization time, clearly. Axioms CSIA1-CSIA8 (Table 2) closely resemble the axioms for initial abstraction of processes. Axioms SCG1, SCG2ID, SASGC1 and SASGC2 from Table 3 are the crucial axioms of conditionals. Axioms SCG1, SCG2ID and SASGC1 reflect the informal explanation of the conditional operator given above. Axiom SASGC2, also called the *time spectrum expansion* axiom, indicates that a parametric time process can be regarded as including a separate alternative for each initialization time. These alternatives are expressed by terms of the form $\mathbf{pt}(v) :: \rightarrow \bar{v}_{\text{abs}}^v(x)$. The important point here is that $\bar{v}_{\text{abs}}^v(x)$ is a process with absolute timing, i.e. it can be written with the constants and operators of $\text{ACP}^{\text{sat}}\text{I}$ only.

Examples Using the time spectrum expansion axiom, we can derive:

$$\begin{aligned} \sqrt{s}v . (\sigma_{\text{abs}}^{v+1.2}(\tilde{a}) \parallel \sigma_{\text{abs}}^{3.7}(\tilde{b})) &= \int_{v \in [0, 2.5]} (\mathbf{pt}(v) :: \rightarrow \sigma_{\text{abs}}^{v+1.2}(\tilde{a}) \cdot \sigma_{\text{abs}}^{3.7}(\tilde{b})) + \\ &\int_{v \in [2.5, \infty)} (\mathbf{pt}(v) :: \rightarrow \sigma_{\text{abs}}^{3.7}(\tilde{b}) \cdot \sigma_{\text{abs}}^{v+1.2}(\tilde{a})) + \\ &(\mathbf{pt}(2.5) :: \rightarrow \sigma_{\text{abs}}^{3.7}(\tilde{a} \mid \tilde{b})) \end{aligned}$$

It is easy to check that Lemmas 1 and 2 from Section 2.1 go through for the extension with conditionals.

Semantics First of all, we need the structural operational semantics of $\text{ACP}^{\text{sat}}\text{IV}$ extended with a restricted form of conditionals, viz. conditionals where the condition

| | |
|--|---------|
| $\mathbf{t} :: \rightarrow x = x$ | SGC1 |
| $\mathbf{f} :: \rightarrow x = \dot{\delta}$ | SGC2ID |
| $\bar{v}_{\text{abs}}^p(b :: \rightarrow x) = (\bar{v}_{\text{abs}}^p(b) :: \rightarrow \bar{v}_{\text{abs}}^p(x)) + \sigma_{\text{abs}}^p(\dot{\delta})$ | SASGC1 |
| $x = (\int_{v \in [0, p]} (\text{pt}(v) :: \rightarrow \bar{v}_{\text{abs}}^v(x))) + (\text{pt}_{>}(p) :: \rightarrow x)$ | SASGC2 |
| $b :: \rightarrow \dot{\delta} = \dot{\delta}$ | SGC3ID |
| $(b :: \rightarrow \sigma_{\text{abs}}^p(x)) + \sigma_{\text{abs}}^p(\dot{\delta}) = \sqrt{v} \cdot \sigma_{\text{abs}}^p(\bar{v}_{\text{abs}}^v(b) :: \rightarrow x)$ | SASGC3 |
| $b :: \rightarrow (x + y) = (b :: \rightarrow x) + (b :: \rightarrow y)$ | SGC4 |
| $b :: \rightarrow (x \cdot y) = (b :: \rightarrow x) \cdot y$ | SGC5 |
| $(b \vee b') :: \rightarrow x = (b :: \rightarrow x) + (b' :: \rightarrow x)$ | SGC6 |
| $b :: \rightarrow (b' :: \rightarrow x) = (b \wedge b') :: \rightarrow x$ | SGC7 |
| $b :: \rightarrow v_{\text{abs}}^p(x) = v_{\text{abs}}^p(b :: \rightarrow x)$ | SASGC4 |
| $b :: \rightarrow (x \parallel y) = (b :: \rightarrow x) \parallel (b :: \rightarrow y)$ | SASGC5 |
| $b :: \rightarrow (x \mid y) = (b :: \rightarrow x) \mid (b :: \rightarrow y)$ | SASGC6 |
| $b :: \rightarrow \partial_H(x) = \partial_H(b :: \rightarrow x)$ | SASGC7 |
| $b :: \rightarrow \nu_{\text{abs}}(x) = \nu_{\text{abs}}(b :: \rightarrow x)$ | SASGC8 |
| $D :: \rightarrow (\int_{v \in V} P) = \int_{v \in V} (D :: \rightarrow P)$ | SASGC9 |
| $D :: \rightarrow (\sqrt{v} \cdot F) = \sqrt{v} \cdot (\bar{v}_{\text{abs}}^v(D) :: \rightarrow F)$ | SASGC10 |
| $(\sqrt{v} \cdot C) :: \rightarrow G = \sqrt{v} \cdot (C :: \rightarrow \bar{v}_{\text{abs}}^v(G))$ | SASGC11 |

Table 3: Axioms for conditionals ($p \geq 0$, v not free in D and G)

is either \mathbf{t} or \mathbf{f} . It is described by the rules for ACPSatIV and the rules given in Table 4. On the basis of these rules the conditional operator can also be directly defined on real time transition systems in a straightforward way. In Table 5, the conditional operator is defined on $\text{RTTS}^*(\mathbb{A})$ in terms of this operator. Additionally, the operators introduced for conditions are defined on \mathbb{B}^* . We use c, d, \dots to denote elements of \mathbb{B}^* . As in the case of ACPSatIV , we obtain a model by defining all operators on bisimulation equivalence classes.

| | |
|---|---|
| $\frac{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}{\langle \mathbf{t} :: \rightarrow x, p \rangle \xrightarrow{a} \langle x', p \rangle}$ | $\frac{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{v}, p \rangle}{\langle \mathbf{t} :: \rightarrow x, p \rangle \xrightarrow{a} \langle \sqrt{v}, p \rangle}$ |
| $\frac{\langle x, p \rangle \xrightarrow{r} \langle x, p + r \rangle}{\langle \mathbf{t} :: \rightarrow x, p \rangle \xrightarrow{r} \langle \mathbf{t} :: \rightarrow x, p + r \rangle}$ | $\frac{\text{ID}(x, p)}{\text{ID}(\mathbf{t} :: \rightarrow x, p)}$ $\frac{}{\text{ID}(\mathbf{f} :: \rightarrow x, p)}$ |

Table 4: Rules for conditionals ($a \in \mathbb{A}$, $r > 0$, $p \geq 0$)

| | |
|---|---|
| $c :: \rightarrow f = \lambda t . (c(t) :: \rightarrow f(t))$ | $\neg c = \lambda t . \neg(c(t))$ |
| $\mathbf{t} = \lambda t . \mathbf{t}$ | $c \wedge d = \lambda t . (c(t) \wedge d(t))$ |
| $\mathbf{f} = \lambda t . \mathbf{f}$ | $c \vee d = \lambda t . (c(t) \vee d(t))$ |
| $\text{pt}(s) = \lambda t . (\text{if } t = s \text{ then } \mathbf{t} \text{ else } \mathbf{f})$ | $\bar{v}_{\text{abs}}^p(c) = c(p)$ |
| $\text{pt}_{>}(s) = \lambda t . (\text{if } t > s \text{ then } \mathbf{t} \text{ else } \mathbf{f})$ | $\sqrt{s}^* \gamma = \lambda t . \bar{v}_{\text{abs}}^t(\gamma(t))$ |

Table 5: Definition of conditional operator on RTTS^* ($p \in \mathbb{R}_{\geq 0}$, $s \in \mathbb{R}$, $\gamma : \mathbb{R}_{\geq 0} \rightarrow \mathbb{B}^*$)

Standard initialization axioms The following equation concerning initialization of conditions holds in the model described above, and is derivable for closed terms of sort \mathbb{B}^* :

$$\bar{v}_{\text{abs}}^p(\bar{v}_{\text{abs}}^q(b)) = \bar{v}_{\text{abs}}^q(b) \quad \text{SI18}$$

We will use this axiom in proofs in subsequent sections.

3.2 Absolute timing

Conditions of the forms $\text{pt}(p)$ and $\text{pt}_{>}(p)$ make it possible to express time-dependent conditions without using initial abstraction. As a result, an extension of $\text{ACP}^{\text{sat}}\text{I}$ similar to the extension of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ described in Section 3.1 is possible. This would not have been the case if we had taken conditions of the forms $p = q$ and $p < q$ as basic conditions instead.

The signature and axioms of this extension of $\text{ACP}^{\text{sat}}\text{I}$, called $\text{ACP}^{\text{sat}}\text{IC}$, are as follows. The signature of $\text{ACP}^{\text{sat}}\text{IC}$ is simply the signature of $\text{ACP}^{\text{sat}}\text{I}\checkmark\text{C}$ without the initial abstraction operators for conditions and processes. The axioms of $\text{ACP}^{\text{sat}}\text{IC}$ consists of the axioms of $\text{ACP}^{\text{sat}}\text{I}$, the equations given in Tables 2 and 3 except SASGC3, SASGC10 and SASGC11, and the following equation:

$$\bar{v}_{\text{abs}}^p((b :: \rightarrow \sigma_{\text{abs}}^q(x)) + \sigma_{\text{abs}}^q(\delta)) = \bar{v}_{\text{abs}}^p(\sigma_{\text{abs}}^q(\bar{v}_{\text{abs}}^p(b) :: \rightarrow x)) \quad \text{SASGC3'}$$

Note that axiom SASGC3 can be replaced by axiom SASGC3' in $\text{ACP}^{\text{sat}}\text{I}\checkmark\text{C}$ as well; it follows immediately from axiom SIA5.

We treated $\text{ACP}^{\text{sat}}\text{I}\checkmark\text{C}$ first, despite the fact that it is a conservative extension of $\text{ACP}^{\text{sat}}\text{IC}$. The reason is that semantically the conditionals with time-dependent conditions are simpler to deal with in case of $\text{ACP}^{\text{sat}}\text{I}\checkmark\text{C}$. A model of $\text{ACP}^{\text{sat}}\text{IC}$ can be obtained from the model of $\text{ACP}^{\text{sat}}\text{I}\checkmark\text{C}$ presented in Section 3.1 by taking the subalgebra of bisimulation equivalence classes of elements $f \in \text{RTTS}^*(\mathbf{A})$ such that $\bar{v}_{\text{abs}}^0(f) = f$. An isomorphic model can be obtained by using the variant of real time transition systems described below.

A *real time transition system with initialization times* over \mathbf{A} consists of a set of *states* \mathbf{S} , a *root state* $\rho \in \mathbf{S}$ and four kinds of relations on states:

- a binary relation $\langle -, p \rangle \xrightarrow{a}_{p'} \langle -, p \rangle$ for each $a \in \mathbf{A}$, $p, p' \in \mathbb{R}_{\geq 0}$ where $p' \leq p$,
- a unary relation $\langle -, p \rangle \xrightarrow{a}_{p'} \langle \surd, p \rangle$ for each $a \in \mathbf{A}$, $p, p' \in \mathbb{R}_{\geq 0}$ where $p' \leq p$,
- a binary relation $\langle -, p \rangle \xrightarrow{r}_{p'} \langle -, q \rangle$ for each $r \in \mathbb{R}_{> 0}$, $p, p', q \in \mathbb{R}_{\geq 0}$
where $q = p + r$ and $p' \leq p$,
- a unary relation $\text{ID}_{p'}(-, p)$ for each $p, p' \in \mathbb{R}_{\geq 0}$ where $p' \leq p$;

satisfying

1. if $\langle s, p \rangle \xrightarrow{r+r'}_{p'} \langle s', q \rangle$, $r, r' > 0$, then there is a s'' such that $\langle s, p \rangle \xrightarrow{r}_{p'} \langle s'', p+r \rangle$
and $\langle s'', p+r \rangle \xrightarrow{r'}_{p'} \langle s', q \rangle$;
2. if $\langle s, p \rangle \xrightarrow{r}_{p'} \langle s'', p+r \rangle$ and $\langle s'', p+r \rangle \xrightarrow{r'}_{p'} \langle s', q \rangle$, then $\langle s, p \rangle \xrightarrow{r+r'}_{p'} \langle s', q \rangle$.

We write $\text{RTTS}^+(\mathbf{A})$ for the set of all real time transition systems with initialization times over \mathbf{A} .

We can associate a transition system in $\text{RTTS}^+(\mathbf{A})$ with a closed term t of $\text{ACP}^{\text{sat}}\text{IC}$ like before. The action step, action termination, time step and immediate deadlock relations can be explained by adding the proviso “provided t is initialized at time p' ” to the explanation given for the case of the original real time transition systems in Section 2.1.

The structural operational semantics of $\text{BPA}^{\text{sat}}\text{C}$ is described by the rules given in Tables 6 and 7. In the rules for the conditional operator, use is made of unary relations $p \in [-]$ on conditions (for $p \in \mathbb{R}_{\geq 0}$). In Table 8, these relations are defined using rules in the style of structural operational semantics as well. The intended meaning of $p \in [b]$ is that p belongs to the time points at which condition b holds. Apart from

| | |
|---|--|
| $\frac{\text{ID}_{p'}(\delta, p) \quad \text{ID}_{r'}(\tilde{\delta}, r)}{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ $\frac{\langle \sigma_{\text{abs}}^0(x), p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ $\frac{\langle \sigma_{\text{abs}}^r(x), p+r \rangle \xrightarrow{a}_{p'} \langle \sigma_{\text{abs}}^r(x'), p+r \rangle}{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r_{p'} \langle x, p+r \rangle}{\langle \sigma_{\text{abs}}^q(x), p+q \rangle \vdash^r_{p'} \langle \sigma_{\text{abs}}^q(x), p+q+r \rangle}$ $\frac{q > p}{\langle \sigma_{\text{abs}}^{q+r}(x), p \rangle \vdash^r_{p'} \langle \sigma_{\text{abs}}^{q+r}(x), p+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}{\langle x+y, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle, \langle y+x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r_{p'} \langle x, p+r \rangle}{\langle x+y, p \rangle \vdash^r_{p'} \langle x+y, p+r \rangle, \langle y+x, p \rangle \vdash^r_{p'} \langle y+x, p+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}{\langle x \cdot y, p \rangle \xrightarrow{a}_{p'} \langle x' \cdot y, p \rangle}$ $\frac{\langle x, p \rangle \vdash^r_{p'} \langle x, p+r \rangle}{\langle x \cdot y, p \rangle \vdash^r_{p'} \langle x \cdot y, p+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle, p' \in [b]}{\langle b :: \rightarrow x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r_{p'} \langle x, p+r \rangle, p' \in [b]}{\langle b :: \rightarrow x, p \rangle \vdash^r_{p'} \langle b :: \rightarrow x, p+r \rangle}$ | $\frac{\langle \tilde{a}, 0 \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, 0 \rangle}{\text{ID}_{r'}(\tilde{a}, r)}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}{\langle \sigma_{\text{abs}}^0(x), p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}{\langle \sigma_{\text{abs}}^r(x), p+r \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p+r \rangle}$ $\frac{\text{ID}_{p'}(x, p)}{\text{ID}_{p'}(\sigma_{\text{abs}}^q(x), p+q)}$ $\frac{\neg \text{ID}_0(x, 0)}{\langle \sigma_{\text{abs}}^{q+r}(x), q \rangle \vdash^r_{q'} \langle \sigma_{\text{abs}}^{q+r}(x), q+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}{\langle x+y, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle, \langle y+x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\text{ID}_{p'}(x, p), \text{ID}_{p'}(y, p)}{\text{ID}_{p'}(x+y, p)}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}{\langle x \cdot y, p \rangle \xrightarrow{a}_{p'} \langle y, p \rangle}$ $\frac{\text{ID}_{p'}(x, p)}{\text{ID}_{p'}(x \cdot y, p)}$ $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle, p' \in [b]}{\langle b :: \rightarrow x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\text{ID}_{p'}(x, p), p' \in [b]}{\text{ID}_{p'}(b :: \rightarrow x, p)} \quad \frac{p' \notin [b]}{\text{ID}_{p'}(b :: \rightarrow x, p)}$ |
|---|--|

Table 6: Rules for $\text{BPA}^{\text{sat}}\text{C}$ ($a \in \mathbf{A}$, $r > 0$, $p, p', q, q', r' \geq 0$, $p' \leq p$, $q' \leq q$, $r' \leq r$)

the rules for the initialization operator \bar{v}_{abs} , the rules for the operational semantics of BPA^{sat} (Table 20) have been adapted in a simple uniform way. The additional rules for $\text{ACP}^{\text{sat}}\text{IC}$ are obtained by adapting the additional rules for $\text{ACP}^{\text{sat}}\text{I}$ (Tables 21 and 22) in the same way. Bisimulation on $\text{RTTS}^+(\mathbf{A})$ is defined similar to bisimulation on $\text{RTTS}(\mathbf{A})$. Like before, we obtain a model for $\text{ACP}^{\text{sat}}\text{IC}$ by identifying bisimilar processes.

| | |
|---|--|
| $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle, q > p}{\langle v_{\text{abs}}^q(x), p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ | $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle, q > p}{\langle v_{\text{abs}}^q(x), p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}$ |
| $\frac{\langle x, p \rangle \vdash^r_{p'} \langle x, p+r \rangle, q > p+r}{\langle v_{\text{abs}}^q(x), p \rangle \vdash^r_{p'} \langle v_{\text{abs}}^q(x), p+r \rangle}$ | $\frac{\text{ID}_{p'}(x, p), q > p}{\text{ID}_{p'}(v_{\text{abs}}^q(x), p)}$ |
| $q \leq p$ | |
| $\text{ID}_{p'}(v_{\text{abs}}^q(x), p)$ | |
| $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}{\langle \bar{v}_{\text{abs}}^{p'}(x), p \rangle \xrightarrow{a}_{p'} \langle x', p \rangle}$ | $\frac{\langle x, p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}{\langle \bar{v}_{\text{abs}}^{p'}(x), p \rangle \xrightarrow{a}_{p'} \langle \sqrt{\cdot}, p \rangle}$ |
| $\frac{\langle x, p \rangle \vdash^r_{p'} \langle x, p+r \rangle}{\langle \bar{v}_{\text{abs}}^{p'}(x), p \rangle \vdash^r_{p'} \langle \bar{v}_{\text{abs}}^{p'}(x), p+r \rangle}$ | $\frac{\text{ID}_{p'}(x, p)}{\text{ID}_{p'}(\bar{v}_{\text{abs}}^{p'}(x), p)}$ |
| $q > p$ | |
| $\frac{\langle \bar{v}_{\text{abs}}^{q+r}(x), p \rangle \vdash^r_{p'} \langle \bar{v}_{\text{abs}}^{q+r}(x), p+r \rangle}{\langle \bar{v}_{\text{abs}}^{q+r}(x), q \rangle \vdash^r_{q'} \langle \bar{v}_{\text{abs}}^{q+r}(x), q+r \rangle}$ | $\frac{-\text{ID}_{q'}(x, q+r)}{\langle \bar{v}_{\text{abs}}^{q+r}(x), q \rangle \vdash^r_{q'} \langle \bar{v}_{\text{abs}}^{q+r}(x), q+r \rangle}$ |

Table 7: Rules for $\text{BPA}^{\text{sat}}\text{C}$ ($a \in \mathbf{A}$, $r > 0$, $p, p', q, q' \geq 0$, $p' \leq p$, $q' \leq q$)

| | | | |
|------------------|-------------------------|--|-------------------------------------|
| $p \in [t]$ | $p \in [\text{pt}(p)]$ | $p > s$ | |
| | | $p \in [\text{pt}_{>}(s)]$ | |
| $p \notin [b]$ | $p \in [b], p \in [b']$ | $p \in [b]$ | $q \in [b]$ |
| $p \in [\neg b]$ | $p \in [b \wedge b']$ | $p \in [b \vee b'], p \in [b' \vee b]$ | $p \in [\bar{v}_{\text{abs}}^q(b)]$ |

Table 8: Rules for condition evaluation ($p, q \in \mathbb{R}_{\geq 0}$, $s \in \mathbb{R}$)

3.3 Recursion

In this paper, we do not treat the addition of recursion to any of the presented versions of ACP with timing in detail. However, we describe in this subsection the addition of recursion to $\text{ACPs}^{\text{sat}}\text{IC}$ and $\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$ in outline to make understanding of the specifications given in Section 3.4 easier.

In case of $\text{ACPs}^{\text{sat}}\text{IC}$ and $\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$, recursive specification, solution and guardedness are defined in a similar way as for ACP in [7].

Let V be a set of variables of sort \mathbf{P} (\mathbf{P}^*). A *recursive specification* $E = E(V)$ in $\text{ACPs}^{\text{sat}}\text{IC}$ ($\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$) is a set of equations $E = \{X = t_X \mid X \in V\}$ where each t_X is a $\text{ACPs}^{\text{sat}}\text{IC}$ ($\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$) term that only contains variables from V . A *solution* of a recursive specification $E(V)$ in $\text{ACPs}^{\text{sat}}\text{IC}$ ($\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$) is a set of processes $\{\langle X|E \rangle \mid X \in V\}$ in some model of $\text{ACPs}^{\text{sat}}\text{IC}$ ($\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$) such that the equations of $E(V)$ hold if, for all $X \in V$, X stands for $\langle X|E \rangle$. Mostly, we are interested in one particular variable $X \in V$. When adding recursion, we add constants $\langle X|E \rangle : \rightarrow \mathbf{P}$ ($\langle X|E \rangle : \rightarrow \mathbf{P}^*$) for all recursive specification $E(V)$ and all $X \in V$. For a fixed $E(V)$, the constants $\langle X|E \rangle$ for $X \in V$ make up a solution of $E(V)$.

Let t be a term containing a variable X . We call an occurrence of X in t *guarded* if t has a subterm of the form $\tilde{a} \cdot t'$ or $\sigma_{\text{abs}}^r(t')$ with $r \in \mathbb{R}_{>0}$ and t' a term containing this occurrence of X . We call a recursive specification *guarded* if all occurrences of all its variables in the right-hand sides of all its equations are guarded or it can be rewritten to such a recursive specification using the axioms of $\text{ACPs}^{\text{sat}}\text{IC}$ ($\text{ACPs}^{\text{sat}}\text{I}\vee\text{C}$) and its equations. The Recursive Specification Principle (RSP) states that every guarded

recursive specification has a unique solution. It is possible to obtain a model of $\text{ACP}^{\text{sat}}\text{IC}$ ($\text{ACP}^{\text{sat}}\text{IvC}$) with recursion in which every guarded recursive specification has a unique solution.

Let $E = \{X = t_X \mid X \in V\}$ be a recursive specification in $\text{ACP}^{\text{sat}}\text{IC}$. Then roughly, the additional rules for the operational semantics of $\text{ACP}^{\text{sat}}\text{IC}$ with recursion come down to looking upon $\langle X|E \rangle$ as the process t_X with, for all $X' \in V$, all occurrences of X' in t_X replaced by $\langle X'|E \rangle$. In the model of $\text{ACP}^{\text{sat}}\text{IC}$ with recursion obtained in the same way as for $\text{ACP}^{\text{sat}}\text{IC}$ (Section 3.2), every guarded recursive specification has a unique solution. We obtain a model of $\text{ACP}^{\text{sat}}\text{IvC}$ with recursion in the same way as for $\text{ACP}^{\text{sat}}\text{Iv}$ (Section 3.1). Because of the extensionality of parametric time processes in this model, it is easy to see that in this model every guarded recursive specification in $\text{ACP}^{\text{sat}}\text{IvC}$ has a unique solution.

In the recursive specifications given in Section 3.4, we use equations of the form $X(p) = t$, with p ranging over some interval I of $\mathbb{R}_{\geq 0}$, for a system of equations with one equation for each $p \in I$. The advantage of this view is that the $X(p)$ do not have free variables and no complications arise with name clashes and α -conversion. It is possible to view such equations as single ones instead, but in that case terms with parameters have to be understood in detail.

3.4 Example

We will now use $\text{ACP}^{\text{sat}}\text{IvC}$ in an example concerning railroad crossings. Controlling a railroad crossing involves the behaviour of trains, a gate and a controller. We shall give (guarded recursive) specifications of the behaviour that is relevant to railroad crossing control. We take the following informal description of the time-dependent behaviour of the trains, the gate and the controller from [13] as the starting-point of our specifications. The example originates from [1].

When a train approaches the gate from a great distance its speed is between 48 m/s and 52 m/s . As soon as it passes a detector placed at 1000 m backward from the gate, an *app* signal is sent to the controller. The train may now slow down, but its speed stays between 40 m/s and 52 m/s , and pass the gate. As soon as it passes another detector placed at 100 m forward from the gate, an *exit* signal is sent to the controller. A new train may come after the current one has passed the second detector, but only at a distance greater than or equal to 1500 m . The gate is able to receive *lower* and *raise* signals from the controller at any time. As soon as the gate receives a *lower* signal, it lowers from 90° to 0° at a constant rate of 20° per second. As soon as it receives a *raise* signal, it raises from 0° to 90° at the same rate. The controller is able to receive *app* and *exit* signals from the train detectors at any time. When the controller receives an *app* signal, it takes at most 5 s before a *lower* signal is sent to the gate. When it receives an *exit* signal, it takes at most 5 s before a *raise* signal is sent to the gate. Because of fault tolerance considerations, *app* signals should always cause the gate to go down, and *exit* signals should be ignored while the gate is going down.

In the specifications given below, actions are used to model the acts of sending and receiving signals as well as the acts of passing the gate and completing the opening or the closing of the gate.

$$Trains = \int_{t \in [0, \infty)} (\text{pt}_{\leq}(t - \frac{400}{52}) :: \rightarrow (\sigma_{\text{abs}}^t(\text{app}_{tr}) \cdot Tr_{near}))$$

$$Tr_{near} = \int_{t \in [0, \infty)} ((\text{pt}_{\leq}(t - \frac{1000}{52}) \wedge \text{pt}_{\geq}(t - \frac{1000}{40})) :: \rightarrow (\sigma_{\text{abs}}^t(\text{pass}) \cdot Tr_{past}))$$

$$Tr_{past} = \int_{t \in [0, \infty)} ((\text{pt}_{\leq}(t - \frac{100}{52}) \wedge \text{pt}_{\geq}(t - \frac{100}{40})) :: \rightarrow (\sigma_{\text{abs}}^t(\text{exit}_{tr}) \cdot Trains))$$

Some simple calculations give us the lower and upper bounds for the times at which a train may pass the detectors and the gate. If a train goes at time t_0 from one point to another point at a distance d with a speed between v_l and v_h , then the lower and upper bounds for the time t at which the train passes the latter point are couched by the assertions $t_0 + \frac{d}{v_h} \leq t$ and $t \leq t_0 + \frac{d}{v_l}$, respectively. The conditions used in the specification given above are modelled on the equivalent assertions $t_0 \leq t - \frac{d}{v_h}$ and $t_0 \geq t - \frac{d}{v_l}$. There is only a lower bound in case of the first detector because the train that comes after the current one may be at any distance greater than or equal to 400 m backward from the first detector.

$$Gate = \int_{t \in [0, \infty)} (\sigma_{\text{abs}}^t(\text{lower}_g) \cdot Ga_{dn}(90) + \sigma_{\text{abs}}^t(\text{raise}_g) \cdot Gate)$$

$$Ga_{dn}(a) = \sqrt{s} t_0 \cdot \int_{t \in [0, \infty)} ((\text{pt}(t - \frac{a}{20}) :: \rightarrow (\sigma_{\text{abs}}^t(\text{ready}) \cdot Ga_{cl})) + (\text{pt}_{\geq}(t - \frac{a}{20}) :: \rightarrow (\sigma_{\text{abs}}^t(\text{lower}_g) \cdot Ga_{dn}(a - 20(t - t_0)) + \sigma_{\text{abs}}^t(\text{raise}_g) \cdot Ga_{up}(a - 20(t - t_0))))))$$

$$Ga_{cl} = \int_{t \in [0, \infty)} (\sigma_{\text{abs}}^t(\text{lower}_g) \cdot Ga_{cl} + \sigma_{\text{abs}}^t(\text{raise}_g) \cdot Ga_{up}(0))$$

$$Ga_{up}(a) = \sqrt{s} t_0 \cdot \int_{t \in [0, \infty)} ((\text{pt}(t - \frac{90-a}{20}) :: \rightarrow (\sigma_{\text{abs}}^t(\text{ready}) \cdot Gate)) + (\text{pt}_{\geq}(t - \frac{90-a}{20}) :: \rightarrow (\sigma_{\text{abs}}^t(\text{lower}_g) \cdot Ga_{dn}(a + 20(t - t_0)) + \sigma_{\text{abs}}^t(\text{raise}_g) \cdot Ga_{up}(a + 20(t - t_0))))))$$

where a ranges over the interval $[0, 90]$ of $\mathbb{R}_{\geq 0}$.

While the gate is going up or down, its angle a is relevant whenever a controller signal is received. If that happens, say at time t , the time passed since the previous controller signal was received determines the angle at time t . In the specification of the behaviour of the gate given above, we use initial abstraction to be able to refer to the time at which the previous controller signal was received.

$$Control = \int_{t \in [0, \infty)} (\sigma_{\text{abs}}^t(\text{app}_c) \cdot Co_{dn}(0) + \sigma_{\text{abs}}^t(\text{exit}_c) \cdot Co_{up}(0))$$

$$Co_{dn}(d) = \sqrt{s} t_0 \cdot \int_{t \in [0, \infty)} (\text{pt}_{\geq}(t - (5 - d)) :: \rightarrow (\sigma_{\text{abs}}^t(\text{lower}_c) \cdot Control + \sigma_{\text{abs}}^t(\text{app}_c) \cdot Co_{dn}(d + (t - t_0)) + \sigma_{\text{abs}}^t(\text{exit}_c) \cdot Co_{dn}(d + (t - t_0))))$$

$$Co_{up}(d) = \sqrt{s} t_0 \cdot \int_{t \in [0, \infty)} (\text{pt}_{\geq}(t - (5 - d)) :: \rightarrow (\sigma_{\text{abs}}^t(\text{raise}_c) \cdot Control + \sigma_{\text{abs}}^t(\text{app}_c) \cdot Co_{dn}(0) + \sigma_{\text{abs}}^t(\text{exit}_c) \cdot Co_{up}(d + (t - t_0))))$$

where d ranges over the interval $[0, 5]$ of $\mathbb{R}_{\geq 0}$.

While the controller is preparing for sending a signal to the gate in response to a detector signal, the delay d of the response is relevant whenever another detector signal is received. If that happens, say at time t , the time passed since the previous detector signal was received determines the delay at time t . We use initial abstraction to be able to refer to the time at which the previous detector signal was received.

Let the communication function γ be such that

$$\gamma(\text{app}_{tr}, \text{app}_c) = \text{app}, \gamma(\text{exit}_{tr}, \text{exit}_c) = \text{exit}, \gamma(\text{lower}_c, \text{lower}_g) = \text{lower}, \gamma(\text{raise}_c, \text{raise}_g) = \text{raise}$$

and γ is undefined otherwise. Then the railroad crossing system is described by

where

$$H = \{app_{tr}, app_c, exit_{tr}, exit_c, lower_c, lower_g, raise_c, raise_g\}$$

Analysis of this term can provide answers to various basic questions about the system. It can, for example, be simplified to a term which shows that (1) a train can only pass the gate when the gate is closed, (2) the gate opens after a train has left the track unless a new train has entered the track and (3) the system reacts adequately when a new train enters the track while the gate is going up. We do not give an account of the simplification here. It involves the use of various standard process algebraic techniques, such as linearization of guarded recursive specifications and expansion of parallel composition (see e.g. [13]), of which the treatment in the setting of $\text{ACP}^{\text{sat}}\text{IVC}$ would go beyond the scope of this paper.

4 Discrete time and time-dependent conditions

In this section, we briefly review $\text{ACP}^{\text{dat}}\checkmark$, the discrete time counterpart of $\text{ACP}^{\text{sat}}\text{IV}$ presented in [6], and add a conditional operator with time-dependent conditions to it. In Section 5, we show that the resulting theory, called $\text{ACP}^{\text{dat}}\checkmark\text{C}$, can be embedded in $\text{ACP}^{\text{sat}}\text{IVC}$. In $\text{ACP}^{\text{dat}}\checkmark\text{C}$, the conditions are essentially the same as the conditions introduced earlier in [4]. First, in Section 4.1, we review $\text{ACP}^{\text{dat}}\checkmark$. After that, in Section 4.2, we extend $\text{ACP}^{\text{dat}}\checkmark$ to $\text{ACP}^{\text{dat}}\checkmark\text{C}$.

4.1 Discrete time process algebra

In this subsection, we briefly review ACP^{dat} , a discrete time process algebra with absolute timing, and its extension with initial abstraction. A more detailed account is given in [6]. The axioms – extracted from [6] – are given in Appendix B.

ACP^{dat} is a conservative extensions of ACP_{dat} [4]. In ACP^{dat} , time is measured on a discrete time scale. The discrete time points divide time into time slices and timing of actions is done with respect to the time slices in which they are performed – “in time slice $n + 1$ ” means “at some time point p such that $n \leq p < n + 1$ ”.

In ACP^{dat} , we have the constants \underline{a} and $\underline{\delta}$ instead of \tilde{a} and $\tilde{\delta}$. The constants \underline{a} and $\underline{\delta}$ stand for a in time slice 1 and a deadlock in time slice 1, respectively. The operators σ_{abs} , ν_{abs} and $\bar{\nu}_{\text{abs}}$ have a natural number instead of a non-negative real number as their first argument. The process $\sigma_{\text{abs}}^n(x)$ is the process x shifted in time by n on the discrete time scale. The process $\nu_{\text{abs}}^n(x)$ is the part of x that starts to perform actions before time slice $n + 1$. The process $\bar{\nu}_{\text{abs}}^n(x)$ is the part of x that starts to perform actions in time slice $n + 1$ or a later time slice. Recall that time point n is the starting-point of time slice $n + 1$. In ACP^{dat} , we do not have a discrete time counterpart of ν_{abs} . Unlike before in the case of real time, we can use ν_{abs}^1 instead. The initial abstraction operator \checkmark_{d} is the discrete counterpart of \checkmark_{s} . This means that $\checkmark_{\text{d}}i.F$, where i is a variable ranging over \mathbb{N} and F is a term that may contain free variables, denotes a function $f : \mathbb{N} \rightarrow \mathbf{P}$ that satisfies $f(n) = \bar{\nu}_{\text{abs}}^n(f(n))$ for all $n \in \mathbb{N}$. In the resulting theory, called $\text{ACP}^{\text{dat}}\checkmark$, the sort \mathbf{P} of processes is replaced by the sort \mathbf{P}^* of parametric time processes.

We denote elements of \mathbb{N} by m, m', n, n' . We assume that an infinite set of time variables ranging over \mathbb{N} has been given, and denote them by i, j, \dots . We denote terms of $\text{ACP}^{\text{dat}\checkmark}$ by F, G, \dots .

Axiom systems The axiom system of BPA^{dat} consists of the equations given in Table 24. The axiom system of ACP^{dat} consists of the axioms of BPA^{dat} and the equations given in Table 25. The axiom system of $\text{ACP}^{\text{dat}\checkmark}$ consists of the axioms of ACP^{dat} and the equations given in Table 26. For a discussion of the axioms of BPA^{dat} , ACP^{dat} and $\text{ACP}^{\text{dat}\checkmark}$, see [6].

Semantics In case a discrete time scale is used, we use a variant of real time transition systems, called *discrete time transition systems*, with only relations $\langle -, p \rangle \xrightarrow{a} \langle -, p \rangle$, $\langle -, p \rangle \xrightarrow{a} \langle \surd, p \rangle$, $\langle -, p \rangle \xrightarrow{r} \langle -, q \rangle$ and $\text{ID}(-, p)$ for $p, q \in \mathbb{N}$, $r \in \mathbb{N}_{>0}$. We write $\text{DTTS}(\mathbf{A})$ for the set of all discrete time transition systems over \mathbf{A} . Associating a transition system in $\text{DTTS}(\mathbf{A})$ with a closed term t of BPA^{dat} and ACP^{dat} proceeds in essentially the same way as associating a transition system in $\text{RTTS}(\mathbf{A})$ with a closed term t of BPA^{sat} and ACP^{sat} . The only difference is that in the rules for the operational semantics of BPA^{dat} and ACP^{dat} all numbers involved are restricted to \mathbb{N} . For $\text{ACP}^{\text{dat}\checkmark}$, we have to extend $\text{DTTS}(\mathbf{A})$ to the function space

$$\text{DTTS}^*(\mathbf{A}) = \{f : \mathbb{N} \rightarrow \text{DTTS}(\mathbf{A}) \mid \forall n \in \mathbb{N} \bullet f(n) = \bar{v}_{\text{abs}}^n(f(n))\}$$

4.2 Conditionals with time-dependent conditions

We add a conditional operator with time-dependent conditions to $\text{ACP}^{\text{dat}\checkmark}$. The time-dependent conditions introduced here were originally introduced in [4] (see also [5]).

First of all, we introduce time-dependent conditions for the discrete time case. We have the *in time slice* operator sl and the *in time slice greater than* operator $\text{sl}_{>}$ instead of pt and $\text{pt}_{>}$. The operator \bar{v}_{abs} has a natural number instead of a non-negative real number as its first argument. For a time-dependent condition b , $\bar{v}_{\text{abs}}^n(b)$ is either \mathbf{t} or \mathbf{f} , determined by whether b holds in time slice $n + 1$ or not. For $n \in \mathbb{N}$, the condition $\text{sl}(n)$ holds only in time slice n and the condition $\text{sl}_{>}(n)$ holds in all time slices greater than n . For $m \in \mathbb{N}_{>0}$, the condition $\text{sl}(-m)$ never holds and the condition $\text{sl}_{>}(-m)$ always holds. We also have the initial abstraction operator \surd_{d} , instead of \surd_{s} , for conditions.

We join time-dependent conditions with parametric time processes by means of the conditional operator $::\rightarrow$. In $\text{ACP}^{\text{dat}\checkmark}\mathbf{C}$, we have, in addition to the above-mentioned constants and operators on \mathbb{B}^* , the constants and operators of $\text{ACP}^{\text{dat}\checkmark}$ and the *conditional* operator $::\rightarrow: \mathbb{B}^* \times \mathbf{P}^* \rightarrow \mathbf{P}^*$. Initialized in time slices where the condition b holds, the process $b ::\rightarrow x$ proceeds as the process x ; initialized in other time slices, $b ::\rightarrow x$ is the same as immediate deadlock.

Axiom system The axiom system of $\text{ACP}^{\text{dat}\checkmark}\mathbf{C}$ consists of the axioms of $\text{ACP}^{\text{dat}\checkmark}$ and the equations given in Tables 1, 9 and 10.

Semantics In Table 11, the conditional operator is defined on $\text{DTTS}^*(\mathbf{A})$ in terms of the conditional operator on discrete time transition systems for the conditions \mathbf{t} and \mathbf{f} (see also Section 3.1). Additionally, the operators introduced for conditions are defined on \mathbb{B}^* . In this table, t is a variable ranging over \mathbb{N} . As in the case of $\text{ACP}^{\text{dat}\checkmark}$, we obtain a model by defining all operators on bisimulation equivalence classes.

| | | | |
|---|--------|--|-------|
| $\bar{v}_{\text{abs}}^n(\mathbf{t}) = \mathbf{t}$ | CDAI1 | $\sqrt{d} j . D = \sqrt{d} i . D[i/j]$ | CDIA1 |
| $\bar{v}_{\text{abs}}^n(\mathbf{f}) = \mathbf{f}$ | CDAI2 | $\bar{v}_{\text{abs}}^n(\sqrt{d} i . C) = \bar{v}_{\text{abs}}^n(C[n/i])$ | CDIA2 |
| $\bar{v}_{\text{abs}}^n(\text{sl}(n+1)) = \mathbf{t}$ | CDAI3 | $\sqrt{d} i . (\sqrt{d} j . C) = \sqrt{d} i . C[i/j]$ | CDIA3 |
| $\bar{v}_{\text{abs}}^n(\text{sl}((n+1)-m)) = \mathbf{f}$ | CDAI4 | $D = \sqrt{d} i . D$ | CDIA4 |
| $\bar{v}_{\text{abs}}^n(\text{sl}((n+1)+m)) = \mathbf{f}$ | CDAI5 | $(\forall n \in \mathbb{N} \bullet \bar{v}_{\text{abs}}^n(b) = \bar{v}_{\text{abs}}^n(b')) \Rightarrow b = b'$ | CDIA5 |
| $\bar{v}_{\text{abs}}^n(\text{sl}_{>}((n+1)-m)) = \mathbf{t}$ | CDAI6 | | |
| $\bar{v}_{\text{abs}}^n(\text{sl}_{>}((n+1)+n')) = \mathbf{f}$ | CDAI7 | $\neg(\sqrt{d} i . C) = \sqrt{d} i . \neg C$ | CDIA6 |
| $\bar{v}_{\text{abs}}^n(\neg b) = \neg \bar{v}_{\text{abs}}^n(b)$ | CDAI8 | $(\sqrt{d} i . C) \wedge D = \sqrt{d} i . (C \wedge \bar{v}_{\text{abs}}^i(D))$ | CDIA7 |
| $\bar{v}_{\text{abs}}^n(b \wedge b') = \bar{v}_{\text{abs}}^n(b) \wedge \bar{v}_{\text{abs}}^n(b')$ | CDAI9 | $(\sqrt{d} i . C) \vee D = \sqrt{d} i . (C \vee \bar{v}_{\text{abs}}^i(D))$ | CDIA8 |
| $\bar{v}_{\text{abs}}^n(b \vee b') = \bar{v}_{\text{abs}}^n(b) \vee \bar{v}_{\text{abs}}^n(b')$ | CDAI10 | | |

Table 9: Axioms for conditions ($n, n' \geq 0, m > 0, i$ not free in D)

| | |
|--|--------|
| $\mathbf{t} :: \rightarrow x = x$ | SGC1 |
| $\mathbf{f} :: \rightarrow x = \dot{\delta}$ | SGC2ID |
| $\bar{v}_{\text{abs}}^n(b :: \rightarrow x) = (\bar{v}_{\text{abs}}^n(b) :: \rightarrow \bar{v}_{\text{abs}}^n(x)) + \sigma_{\text{abs}}^n(\dot{\delta})$ | DASGC1 |
| $x = (\sum_{k \in [0, n]} (\text{sl}(k+1) :: \rightarrow \bar{v}_{\text{abs}}^k(x))) + (\text{sl}_{>}(n+1) :: \rightarrow x)$ | DASGC2 |
| $b :: \rightarrow \dot{\delta} = \dot{\delta}$ | SGC3ID |
| $(b :: \rightarrow \sigma_{\text{abs}}^n(x)) + \sigma_{\text{abs}}^n(\dot{\delta}) = \sqrt{d} i . \sigma_{\text{abs}}^n(\bar{v}_{\text{abs}}^i(b) :: \rightarrow x)$ | DASGC3 |
| $b :: \rightarrow (x + y) = (b :: \rightarrow x) + (b :: \rightarrow y)$ | SGC4 |
| $b :: \rightarrow (x \cdot y) = (b :: \rightarrow x) \cdot y$ | SGC5 |
| $(b \vee b') :: \rightarrow x = (b :: \rightarrow x) + (b' :: \rightarrow x)$ | SGC6 |
| $b :: \rightarrow (b' :: \rightarrow x) = (b \wedge b') :: \rightarrow x$ | SGC7 |
| $b :: \rightarrow v_{\text{abs}}^n(x) = v_{\text{abs}}^n(b :: \rightarrow x)$ | DASGC4 |
| $b :: \rightarrow (x \parallel y) = (b :: \rightarrow x) \parallel (b :: \rightarrow y)$ | DASGC5 |
| $b :: \rightarrow (x \mid y) = (b :: \rightarrow x) \mid (b :: \rightarrow y)$ | DASGC6 |
| $b :: \rightarrow \partial_H(x) = \partial_H(b :: \rightarrow x)$ | DASGC7 |
| $D :: \rightarrow (\sqrt{d} i . F) = \sqrt{d} i . (\bar{v}_{\text{abs}}^i(D) :: \rightarrow F)$ | DASGC8 |
| $(\sqrt{d} i . C) :: \rightarrow G = \sqrt{d} i . (C :: \rightarrow \bar{v}_{\text{abs}}^i(G))$ | DASGC9 |

Table 10: Axioms for conditionals ($n \geq 0, i$ not free in D and G)

| | |
|---|---|
| $c :: \rightarrow f = \lambda t . (c(t) :: \rightarrow f(t))$ | $\neg c = \lambda t . \neg(c(t))$ |
| $\mathbf{t} = \lambda t . \mathbf{t}$ | $c \wedge d = \lambda t . (c(t) \wedge d(t))$ |
| $\mathbf{f} = \lambda t . \mathbf{f}$ | $c \vee d = \lambda t . (c(t) \vee d(t))$ |
| $\text{sl}(k) = \lambda t . (\text{if } t+1 = k \text{ then } \mathbf{t} \text{ else } \mathbf{f})$ | $\bar{v}_{\text{abs}}^n(c) = c(n)$ |
| $\text{sl}_{>}(k) = \lambda t . (\text{if } t+1 > k \text{ then } \mathbf{t} \text{ else } \mathbf{f})$ | $\sqrt{s}^* \gamma = \lambda t . \bar{v}_{\text{abs}}^t(\gamma(t))$ |

Table 11: Definition of conditional operator on DTTS* ($n \in \mathbb{N}, k \in \mathbb{Z}, \gamma : \mathbb{N} \rightarrow \mathbb{B}^*$)

5 Embedding

In this section, we will show that $\text{ACP}^{\text{dat}} \vee \text{C}$ can be embedded in $\text{ACP}^{\text{sat}} \text{I} \vee \text{C}$. We will establish the existence of an embedding as follows. We give explicit definitions of the constants and operators in the signature of $\text{ACP}^{\text{dat}} \vee \text{C}$ that are not in the signature of $\text{ACP}^{\text{sat}} \text{I} \vee \text{C}$ and we prove that for closed terms the axioms of $\text{ACP}^{\text{dat}} \vee \text{C}$ are derivable from the axioms of $\text{ACP}^{\text{sat}} \text{I} \vee \text{C}$ and the explicit definitions. The soundness of this

method is discussed in [6]. The explicit definitions needed are given in Table 12.

| | |
|---|--|
| $\underline{a} = \int_{v \in [0,1]} \sigma_{\text{abs}}^v(\tilde{a})$ | $\text{sl}(k) = \text{pt}_{\geq}(k-1) \wedge \text{pt}_{<}(k)$ |
| $\sigma_{\text{abs}}^n(x) = \sigma_{\text{abs}}^n(x)$ | $\text{sl}_{>}(k) = \text{pt}_{\geq}(k)$ |
| $v_{\text{abs}}^n(x) = v_{\text{abs}}^n(x)$ | $\sqrt{d} i . C = \sqrt{v} . C[[v]/i]$ |
| $\bar{v}_{\text{abs}}^n(x) = \bar{v}_{\text{abs}}^n(x)$ | |
| $\sqrt{d} i . F = \sqrt{v} . F[[v]/i]$ | |

Table 12: Definitions of discrete time operators ($a \in \mathbf{A}_\delta$, $n \in \mathbb{N}$, $k \in \mathbb{Z}$)

Before we establish the existence of an embedding, we first take another look at the connection between ACPsatIvC and ACPdatvC by introducing the notion of a discretized real time process. Discrete time processes can be viewed as real time processes that are discretized. We define the notion of a discretized real time process in terms of the auxiliary *discretization* operators $\mathcal{D} : \mathbf{P}^* \rightarrow \mathbf{P}^*$ and $\mathcal{D} : \mathbb{B}^* \rightarrow \mathbb{B}^*$ of which the defining axioms are given in Table 13. In [6], discretization is also defined

| | |
|---|---|
| $\mathcal{D}(\dot{\delta}) = \dot{\delta}$ | $\mathcal{D}(t) = t$ |
| $\mathcal{D}(\tilde{a}) = \underline{a}$ | $\mathcal{D}(f) = f$ |
| $\mathcal{D}(\sigma_{\text{abs}}^p(x)) = \sigma_{\text{abs}}^{\lfloor p \rfloor}(\mathcal{D}(x))$ | $\mathcal{D}(\text{pt}(s)) = \text{sl}(\lfloor s + 1 \rfloor)$ |
| $\mathcal{D}(x + y) = \mathcal{D}(x) + \mathcal{D}(y)$ | $\mathcal{D}(\text{pt}_{>}(s)) = \text{sl}_{>}(\lfloor s \rfloor)$ |
| $\mathcal{D}(x \cdot y) = \mathcal{D}(x) \cdot \mathcal{D}(y)$ | $\mathcal{D}(\neg b) = \neg \mathcal{D}(b)$ |
| $\mathcal{D}(b :: \rightarrow x) = \mathcal{D}(b) :: \rightarrow \mathcal{D}(x)$ | $\mathcal{D}(b \wedge b') = \mathcal{D}(b) \wedge \mathcal{D}(b')$ |
| $\mathcal{D}(\int_{v \in V} F) = \int_{v \in V} \mathcal{D}(F)$ | $\mathcal{D}(b \vee b') = \mathcal{D}(b) \vee \mathcal{D}(b')$ |
| $\mathcal{D}(\sqrt{v} . F) = \sqrt{v} . \mathcal{D}(F)$ | $\mathcal{D}(\bar{v}_{\text{abs}}^p(b)) = \bar{v}_{\text{abs}}^{\lfloor p \rfloor}(\mathcal{D}(b))$ |
| | $\mathcal{D}(\sqrt{v} . C) = \sqrt{v} . \mathcal{D}(C)$ |

Table 13: Definition of discretization ($a \in \mathbf{A}_\delta$, $p \in \mathbb{R}_{\geq 0}$, $s \in \mathbb{R}$)

on the domain of the model of ACPsatIvC from Section 3.1. A real time process x is a *discretized* real time process, written $x \in \text{DIS}$, if $x = \mathcal{D}(x)$. The notion of a discretized real time condition is defined in the same way. The relevant closure properties of discretized real time processes and discretized real time conditions are given in Table 14. Hence, restriction of the domain of the model of ACPsatIvC to the

| | |
|---|---|
| $\dot{\delta}, \underline{a} \in \text{DIS}$ | $t, f, \text{sl}(k), \text{sl}_{>}(k) \in \text{DIS}$ |
| $x \in \text{DIS} \Rightarrow \sigma_{\text{abs}}^n(x), v_{\text{abs}}^n(x), \bar{v}_{\text{abs}}^n(x), \partial_H(x) \in \text{DIS}$ | $b \in \text{DIS} \Rightarrow \neg b, \bar{v}_{\text{abs}}^n(b) \in \text{DIS}$ |
| $x, y \in \text{DIS} \Rightarrow x + y, x \cdot y, x \parallel y, x \parallel\!\!\parallel y, x \mid y \in \text{DIS}$ | $b, b' \in \text{DIS} \Rightarrow b \wedge b', b \vee b' \in \text{DIS}$ |
| $b \in \text{DIS}, x \in \text{DIS} \Rightarrow b :: \rightarrow x \in \text{DIS}$ | $(\forall n \in \mathbb{N} \bullet C[n/i] \in \text{DIS}) \Rightarrow$ |
| $(\forall n \in \mathbb{N} \bullet F[n/i] \in \text{DIS}) \Rightarrow \sqrt{d} i . F \in \text{DIS}$ | $\sqrt{d} i . C \in \text{DIS}$ |
| $(\forall p \in V \bullet F[p/v] \in \text{DIS}) \Rightarrow \int_{v \in V} F \in \text{DIS}$ | |
| $x \in \text{DIS} \Rightarrow \mathcal{D}(x) \in \text{DIS}$ | $b \in \text{DIS} \Rightarrow \mathcal{D}(b) \in \text{DIS}$ |

Table 14: Properties of discretized processes and conditions ($a \in \mathbf{A}_\delta$, $n \in \mathbb{N}$, $k \in \mathbb{Z}$)

discretized elements yields a subalgebra of that model. Because we will prove that for closed terms the axioms of ACPdatvC are derivable from the axioms of ACPsatIvC and the explicit definitions, this subalgebra induces a model of ACPdatvC .

The following lemmas present other useful properties of discrete time processes. These lemmas are used to shorten the calculations in the proof of Theorem 6.

Lemma 3 *In $\text{ACP}^{\text{sat}}\text{I}\check{V}\text{C}$:*

1. *for each closed term b of sort \mathbb{B}^* generated by the embedded constants and operators of $\text{ACP}^{\text{dat}}\check{V}\text{C}$, $b = \sqrt{s} v \cdot \bar{v}_{\text{abs}}^{[v]}(b)$;*
2. *for each closed term t of sort \mathbb{P}^* generated by the embedded constants and operators of $\text{ACP}^{\text{dat}}\check{V}\text{C}$, $t = \sqrt{s} v \cdot \bar{v}_{\text{abs}}^{[v]}(t)$.*

Lemma 4 *For each $p \in \mathbb{R}_{\geq 0}$ and closed term t of $\text{ACP}^{\text{sat}}\text{I}\check{V}\text{C}$ generated by the embedded constants and operators of $\text{ACP}^{\text{dat}}\check{V}\text{C}$, there exists a closed term t' such that $\bar{v}_{\text{abs}}^p(t) = \sigma_{\text{abs}}^p(t')$, $t' = \bar{v}_{\text{abs}}^0(t')$, and if $p \in [0, 1)$ and $\bar{v}_{\text{abs}}^p(t) \neq \sigma_{\text{abs}}^p(\dot{\delta})$, $t' = t' + \sigma_{\text{abs}}^{1-p}(\dot{\delta})$ and $\bar{v}_{\text{abs}}^p(t + \underline{\delta}) = \sigma_{\text{abs}}^p(t' + \tilde{\delta})$.*

Lemma 5 *For each closed term t of $\text{ACP}^{\text{sat}}\text{I}\check{V}\text{C}$ generated by the embedded constants and operators of $\text{ACP}^{\text{dat}}\check{V}\text{C}$, there exists a term t' containing no other free variable than w such that $v_{\text{abs}}^1(t + \underline{\delta}) = \sqrt{s} w \cdot \int_{v \in [0, 1)} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t') + \tilde{\delta})$.*

Lemmas 3.2, 4 and 5 are lemmas 7, 9 and 10, respectively, from [6] adapted to the case with conditionals. It suffices to extend the proofs of those lemmas with the case that t is of the form $b ::= \rightarrow t'$. This is outlined in Appendix C.

Lemma 3 points out that for a real time process corresponding to a discrete time process, the initialization time can always be taken to be a discrete point in time. Lemma 4 shows that for a real time process corresponding to a discrete time process, and for $p \in [0, 1)$ such that the whole process is able to reach time p , the part of the process that starts to perform actions at time p or later is able to reach any time $q \in [p, 1)$. Lemma 5 indicates that for a real time process corresponding to a discrete time process, the part of the process that starts to perform actions before time 1 can be regarded as a real time process that starts to perform actions at time 0 shifted in time by any $p \in [0, 1)$ – and parametrized by the initialization time of the whole process.

The existence of an embedding of $\text{ACP}^{\text{dat}}\check{V}\text{C}$ in $\text{ACP}^{\text{sat}}\text{I}\check{V}\text{C}$ is now established by proving the following theorem.

Theorem 6 (Embedding $\text{ACP}^{\text{dat}}\check{V}\text{C}$ in $\text{ACP}^{\text{sat}}\text{I}\check{V}\text{C}$) *For closed terms, the axioms of $\text{ACP}^{\text{dat}}\check{V}\text{C}$ are derivable from the axioms of $\text{ACP}^{\text{sat}}\text{I}\check{V}\text{C}$ and the explicit definitions of the constants and operators \underline{a} , σ_{abs} , v_{abs} , \bar{v}_{abs} , \sqrt{d} (for processes as well as conditions), sl and $\text{sl}_>$ in Table 12.*

This is Theorem 12 from [6] adapted to the case with conditionals. Because some lemmas used in the proof of that theorem had to be adapted to the case with conditionals as well, minor changes to the proofs for some axioms of $\text{ACP}^{\text{dat}}\check{V}$ are needed. What remains to be shown is that the additional axioms for conditionals are derivable for closed terms. This is outlined in Appendix C.

6 Concluding remarks

We extended the main real time version of ACP presented in [6] with conditionals in which the condition depends on time. We illustrated how this extension can be

used by means of an example concerning a simple hybrid system, namely a railroad crossing system. We also extended the main discrete time version of ACP presented in [6] with conditionals in which the condition depends on time. The conditions introduced in this case are essentially the same as the ones originally introduced in [4]. We demonstrated that the presented real time version of ACP with time-dependent conditions and conditionals generalizes the presented discrete time version of ACP with time-dependent conditions and conditionals.

The discrete time version of ACP with time-dependent conditions and conditionals presented in [4] can not be embedded in the one presented here – although the conditions introduced are essentially the same. The reason is that one of the auxiliary operators used in [4] for the axiomatization of the time-dependent conditions and conditionals, viz. the spectrum tail operator μ , can not be explicitly defined in the version presented here. We refrained from introducing an additional operator making this operator explicitly definable because its usefulness in practice remains doubtful.

In Section 5, we introduced the discretization operator to define the notion of a discretized real time process. However, this is not the only application of this operator. Having a closed term t denoting some real time process, one often obtains by apposite change of the time scale a closed term t' denoting a discretized real time process, i.e. $t' = \mathcal{D}(t)$. In that case, the process can safely be considered at a more abstract level where time is measured with finite precision, i.e. on a discrete time scale. This means that analysis of the real time process t can be replaced by analysis of the discrete time process $\mathcal{D}(t)$. The point here is that the abstraction made in the discrete time case makes processes better amenable to analysis.

It is frequently useful to abstract fully from the timing aspects of a process at a certain stage of its analysis. This is, for example, the case in the analysis of a railroad crossing system outlined in Section 3.4. Further extension of the real time and discrete time versions of ACP presented in this paper with time abstraction appears to be important to make them suitable for being applied in a fully formal way.

References

- [1] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22:181–201, 1996.
- [2] J.C.M. Baeten and J.A. Bergstra. Real time process algebra. *Formal Aspects of Computing*, 3(2):142–188, 1991.
- [3] J.C.M. Baeten and J.A. Bergstra. Real space process algebra. *Formal Aspects of Computing*, 5(6):481–529, 1993.
- [4] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. *Formal Aspects of Computing*, 8(2):188–208, 1996.
- [5] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra: Absolute time, relative time and parametric time. *Fundamenta Informaticae*, 29(1/2):51–76, 1997.

- [6] J.C.M. Baeten and C.A. Middelburg. Process algebra with timing: Real time and discrete time. Computing Science Report 99-11, Eindhoven University of Technology, Department of Mathematics and Computing Science, September 1999. To appear in J.A. Bergstra, A. Ponse and S.A. Smolka, editors, Handbook of Process Algebra, Elsevier, 2000.
- [7] J.C.M. Baeten and C. Verhoef. Concrete process algebra. In S. Abramsky, D. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume IV*, pages 149–268. Oxford University Press, 1995.
- [8] J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.
- [9] J.A. Bergstra and J.W. Klop. The algebra of recursively defined processes and the algebra of regular processes. In *Proceedings 11th ICALP*, pages 82–95. LNCS 172, Springer Verlag, 1984.
- [10] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31:560–599, 1984.
- [11] L. Chen. An interleaving model for real-time systems. In A. Nerode and M. Taitlin, editors, *Symposium on Logical Foundations of Computer Science*, pages 81–92. LNCS 620, Springer-Verlag, 1992.
- [12] W.J. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Information and Computation*, 146:24–54, 1998.
- [13] J.F. Groote and J.J. van Wamel. Analysis of three hybrid systems in timed μ CRL. CWI Report SEN-R9815, Centre for Mathematics and Computer Science, September 1998.
- [14] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
- [15] A.S. Klusener. Completeness in real-time process algebra. In J.C.M. Baeten and J.F. Groote, editors, *CONCUR'91*, pages 376–392. LNCS 527, Springer-Verlag, 1991.
- [16] R. Milner. *A Calculus of Communicating Systems*. LNCS 92, Springer-Verlag, 1980.
- [17] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- [18] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR'90*, pages 401–415. LNCS 458, Springer-Verlag, 1990.
- [19] X. Nicollin and J. Sifakis. The algebra of timed processes ATP: Theory and application. *Information and Computation*, 114:131–178, 1994.
- [20] G.D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, University of Aarhus, Department of Computer Science, 1981.

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

A ACP^{sat} , integration and initial abstraction

Axiom systems The axiom system of BPA^{sat} consists of the equations given in Table 15. The axiom system of ACP^{sat} consists of the equations given in Tables 15 and 16. The axioms for integration are given in Table 17 and the axioms for standard initial abstraction are given in Table 18. In Table 19, some equations concerning initialization and time-out are given that hold in the model \mathbf{M}_A^* described in Section 2.3, and that are derivable for closed terms of $\text{ACP}^{\text{sat}}\text{I}\checkmark$.

Semantics The structural operational semantics of BPA^{sat} is described by the rules given in Table 20. The structural operational semantics of ACP^{sat} is described by the rules given in Tables 20 and 21. The rules for integration are given in Table 22.

In Table 23, the constants and operators of $\text{ACP}^{\text{sat}}\text{I}\checkmark$ are defined on $\text{RTTS}^*(\mathbf{A})$. We use f, g, \dots to denote elements of $\text{RTTS}^*(\mathbf{A})$. We use λ -notation for functions, t is a variable ranging over $\mathbb{R}_{\geq 0}$. We write $f(t) * g$ for the real time transition system obtained from $f(t)$ by replacing $\langle s, p \rangle \xrightarrow{a} \langle \surd, p \rangle$ by $\langle s, p \rangle \xrightarrow{a} \langle s', p \rangle$, where s' is the root state of $g(p)$, whenever s is reachable from the root state of $f(t)$.

| | | | |
|--|-------|---|-------|
| $x + y = y + x$ | A1 | $\tilde{\delta} \cdot x = \tilde{\delta}$ | A7SA |
| $(x + y) + z = x + (y + z)$ | A2 | | |
| $x + x = x$ | A3 | $v_{\text{abs}}^p(\dot{\delta}) = \dot{\delta}$ | SATO0 |
| $(x + y) \cdot z = (x \cdot z) + (y \cdot z)$ | A4 | $v_{\text{abs}}^0(x) = \dot{\delta}$ | SATO1 |
| $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ | A5 | $v_{\text{abs}}^r(\tilde{a}) = \tilde{a}$ | SATO2 |
| $x + \dot{\delta} = x$ | A6ID | $v_{\text{abs}}^{p+q}(\sigma_{\text{abs}}^p(x)) = \sigma_{\text{abs}}^p(v_{\text{abs}}^q(x))$ | SATO3 |
| $\dot{\delta} \cdot x = \dot{\delta}$ | A7ID | $v_{\text{abs}}^p(x + y) = v_{\text{abs}}^p(x) + v_{\text{abs}}^p(y)$ | SATO4 |
| | | $v_{\text{abs}}^p(x \cdot y) = v_{\text{abs}}^p(x) \cdot y$ | SATO5 |
| $\sigma_{\text{abs}}^0(x) = \bar{v}_{\text{abs}}^0(x)$ | SAT1 | | |
| $\sigma_{\text{abs}}^p(\sigma_{\text{abs}}^q(x)) = \sigma_{\text{abs}}^{p+q}(x)$ | SAT2 | $\bar{v}_{\text{abs}}^0(\dot{\delta}) = \dot{\delta}$ | SAI0a |
| $\sigma_{\text{abs}}^p(x) + \sigma_{\text{abs}}^p(y) = \sigma_{\text{abs}}^p(x + y)$ | SAT3 | $\bar{v}_{\text{abs}}^r(\dot{\delta}) = \sigma_{\text{abs}}^r(\dot{\delta})$ | SAI0b |
| $\sigma_{\text{abs}}^p(x) \cdot v_{\text{abs}}^p(y) = \sigma_{\text{abs}}^p(x \cdot \dot{\delta})$ | SAT4 | $\bar{v}_{\text{abs}}^0(\tilde{a}) = \tilde{a}$ | SAI1 |
| $\sigma_{\text{abs}}^p(x) \cdot (v_{\text{abs}}^p(y) + \sigma_{\text{abs}}^p(z)) = \sigma_{\text{abs}}^p(x \cdot \bar{v}_{\text{abs}}^0(z))$ | SAT5 | $\bar{v}_{\text{abs}}^r(\tilde{a}) = \sigma_{\text{abs}}^r(\dot{\delta})$ | SAI2 |
| $\sigma_{\text{abs}}^p(\dot{\delta}) \cdot x = \sigma_{\text{abs}}^p(\dot{\delta})$ | SAT6 | $\bar{v}_{\text{abs}}^{p+q}(\sigma_{\text{abs}}^p(x)) = \sigma_{\text{abs}}^p(\bar{v}_{\text{abs}}^q(\bar{v}_{\text{abs}}^0(x)))$ | SAI3 |
| $\tilde{a} + \tilde{\delta} = \tilde{a}$ | A6SAa | $\bar{v}_{\text{abs}}^p(x + y) = \bar{v}_{\text{abs}}^p(x) + \bar{v}_{\text{abs}}^p(y)$ | SAI4 |
| $\sigma_{\text{abs}}^r(x) + \tilde{\delta} = \sigma_{\text{abs}}^r(x)$ | A6SAb | $\bar{v}_{\text{abs}}^p(x \cdot y) = \bar{v}_{\text{abs}}^p(x) \cdot y$ | SAI5 |

Table 15: Axioms of BPA^{sat} ($a \in \mathbf{A}_\delta$, $p, q \geq 0$, $r > 0$)

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

Table 16: Additional axioms for ACP^{sat} ($a, b \in \mathbf{A}_\delta$, $c \in \mathbf{A}$, $p \geq 0$, $r > 0$)

| | | | |
|--|------|---|-------|
| $\int_{w \in V} R = \int_{v \in V} R[v/w]$ | INT1 | $\int_{v \in V} \sigma_{\text{abs}}^p(P) = \sigma_{\text{abs}}^p(\int_{v \in V} P)$ if $p \neq v$ | INT10 |
| $\int_{v \in \emptyset} P = \dot{\delta}$ | INT2 | $\int_{v \in V} (P + Q) = \int_{v \in V} P + \int_{v \in V} Q$ | INT11 |
| $\int_{v \in \{p\}} P = P[p/v]$ | INT3 | $\int_{v \in V} (P \cdot R) = (\int_{v \in V} P) \cdot R$ | INT12 |
| $\int_{v \in V \cup W} P = \int_{v \in V} P + \int_{v \in W} P$ | INT4 | $\int_{v \in V} (P \parallel R) = (\int_{v \in V} P) \parallel R$ | INT13 |
| $V \neq \emptyset \Rightarrow \int_{v \in V} R = R$ | INT5 | $\int_{v \in V} (P \mid R) = (\int_{v \in V} P) \mid R$ | INT14 |
| $(\forall p \in V \bullet P[p/v] = Q[p/v]) \Rightarrow$ | | $\int_{v \in V} (R \mid P) = R \mid (\int_{v \in V} P)$ | INT15 |
| $\int_{v \in V} P = \int_{v \in V} Q$ | INT6 | $\int_{v \in V} \partial_H(P) = \partial_H(\int_{v \in V} P)$ | INT16 |
| $V \neq \emptyset \Rightarrow \int_{v \in V} \sigma_{\text{abs}}^v(\dot{\delta}) = \sigma_{\text{abs}}^{\text{sup } V}(\dot{\delta})$ | INT7 | $\nu_{\text{abs}}^p(\int_{v \in V} P) = \int_{v \in V} \nu_{\text{abs}}^p(P)$ if $p \neq v$ | SAT06 |
| $V \neq \emptyset, \text{sup } V \notin V \Rightarrow$ | | $\bar{\nu}_{\text{abs}}^p(\int_{v \in V} P) = \int_{v \in V} \bar{\nu}_{\text{abs}}^p(P)$ if $p \neq v$ | SAI6 |
| $\int_{v \in V} \sigma_{\text{abs}}^v(\tilde{\delta}) = \sigma_{\text{abs}}^{\text{sup } V}(\tilde{\delta})$ | INT8 | $\nu_{\text{abs}}(\int_{v \in V} P) = \int_{v \in V} \nu_{\text{abs}}(P)$ | SAU5 |
| $\text{sup } V \in V \Rightarrow \int_{v \in V} \sigma_{\text{abs}}^v(\tilde{\delta}) = \sigma_{\text{abs}}^{\text{sup } V}(\tilde{\delta})$ | INT9 | | |

Table 17: Axioms for integration ($p \geq 0$, v not free in R)

| | | | |
|---|------|---|-------|
| $\sqrt{v} \cdot G = \sqrt{v} \cdot G[v/w]$ | SIA1 | $(\sqrt{v} \cdot F) \cdot G = \sqrt{v} \cdot (F \cdot G)$ | SIA9 |
| $\bar{\nu}_{\text{abs}}^p(\sqrt{v} \cdot F) = \bar{\nu}_{\text{abs}}^p(F[p/v])$ | SIA2 | $\nu_{\text{abs}}^p(\sqrt{v} \cdot F) = \sqrt{v} \cdot \nu_{\text{abs}}^p(F)$ if $p \neq v$ | SIA10 |
| $\sqrt{v} \cdot (\sqrt{v} \cdot F) = \sqrt{v} \cdot F[v/w]$ | SIA3 | $(\sqrt{v} \cdot F) \parallel G = \sqrt{v} \cdot (F \parallel \bar{\nu}_{\text{abs}}^v(G))$ | SIA11 |
| $G = \sqrt{v} \cdot G$ | SIA4 | $G \parallel (\sqrt{v} \cdot F) = \sqrt{v} \cdot (\bar{\nu}_{\text{abs}}^v(G) \parallel F)$ | SIA12 |
| $(\forall p \in \mathbb{R}_{\geq 0} \bullet \bar{\nu}_{\text{abs}}^p(x) = \bar{\nu}_{\text{abs}}^p(y)) \Rightarrow$ | | $(\sqrt{v} \cdot F) \mid G = \sqrt{v} \cdot (F \mid \bar{\nu}_{\text{abs}}^v(G))$ | SIA13 |
| $x = y$ | SIA5 | $G \mid (\sqrt{v} \cdot F) = \sqrt{v} \cdot (\bar{\nu}_{\text{abs}}^v(G) \mid F)$ | SIA14 |
| $\sigma_{\text{abs}}^p(\tilde{a}) \cdot x = \sigma_{\text{abs}}^p(\tilde{a}) \cdot \bar{\nu}_{\text{abs}}^p(x)$ | SIA6 | $\partial_H(\sqrt{v} \cdot F) = \sqrt{v} \cdot \partial_H(F)$ | SIA15 |
| $\sigma_{\text{abs}}^p(\sqrt{v} \cdot F) = \sigma_{\text{abs}}^p(F[0/v])$ | SIA7 | $\nu_{\text{abs}}(\sqrt{v} \cdot F) = \sqrt{v} \cdot \nu_{\text{abs}}(F)$ | SIA16 |
| $(\sqrt{v} \cdot F) + G = \sqrt{v} \cdot (F + \bar{\nu}_{\text{abs}}^v(G))$ | SIA8 | $\int_{v \in V} (\sqrt{v} \cdot F) = \sqrt{v} \cdot (\int_{v \in V} F)$ if $v \neq w$ | SIA17 |

Table 18: Axioms for standard initial abstraction ($p \geq 0$, v not free in G)

| | | | |
|--|-----|--|------|
| $\overline{v}_{\text{abs}}^p(v_{\text{abs}}^{p+r}(x)) = v_{\text{abs}}^{p+r}(\overline{v}_{\text{abs}}^p(x))$ | SI1 | $\overline{v}_{\text{abs}}^p(x \parallel y) = \overline{v}_{\text{abs}}^p(x) \parallel \overline{v}_{\text{abs}}^p(y)$ | SI10 |
| $\overline{v}_{\text{abs}}^p(\overline{v}_{\text{abs}}^{p+q}(x)) = \overline{v}_{\text{abs}}^{p+q}(x)$ | SI2 | $\overline{v}_{\text{abs}}^p(x y) = \overline{v}_{\text{abs}}^p(x) \overline{v}_{\text{abs}}^p(y)$ | SI11 |
| $\overline{v}_{\text{abs}}^{p+q}(v_{\text{abs}}^p(x)) = \sigma_{\text{abs}}^{p+q}(\delta)$ | SI3 | $\overline{v}_{\text{abs}}^p(\partial_H(x)) = \partial_H(\overline{v}_{\text{abs}}^p(x))$ | SI12 |
| $v_{\text{abs}}^p(\overline{v}_{\text{abs}}^{p+q}(x)) = \sigma_{\text{abs}}^p(\delta)$ | SI4 | $\overline{v}_{\text{abs}}^0(\nu_{\text{abs}}(x)) = \nu_{\text{abs}}(\overline{v}_{\text{abs}}^0(x))$ | SI13 |
| $\sigma_{\text{abs}}^p(\delta) + \overline{v}_{\text{abs}}^p(x) = \overline{v}_{\text{abs}}^p(x)$ | SI5 | $\overline{v}_{\text{abs}}^r(\nu_{\text{abs}}(x)) = \sigma_{\text{abs}}^r(\delta)$ | SI14 |
| $\sigma_{\text{abs}}^p(\tilde{\delta}) + \overline{v}_{\text{abs}}^p(x + \tilde{\delta}) = \overline{v}_{\text{abs}}^p(x + \tilde{\delta})$ | SI6 | $\nu_{\text{abs}}(\overline{v}_{\text{abs}}^r(x)) = \tilde{\delta}$ | SI15 |
| $\overline{v}_{\text{abs}}^r(x) + \tilde{\delta} = \overline{v}_{\text{abs}}^r(x)$ | SI7 | $v_{\text{abs}}^r(\nu_{\text{abs}}(x)) = \nu_{\text{abs}}(x)$ | SI16 |
| $v_{\text{abs}}^p(v_{\text{abs}}^q(x)) = v_{\text{abs}}^{\min(p,q)}(x)$ | SI8 | $\nu_{\text{abs}}(v_{\text{abs}}^r(x)) = \nu_{\text{abs}}(x)$ | SI17 |
| $\overline{v}_{\text{abs}}^p(\overline{v}_{\text{abs}}^q(\overline{v}_{\text{abs}}^{q'}(x))) = \overline{v}_{\text{abs}}^{\max(p,q)}(\overline{v}_{\text{abs}}^{q'}(x))$ | SI9 | | |

Table 19: Standard initialization axioms ($p, q, q' \geq 0, r > 0$)

| | |
|---|--|
| $\frac{\text{ID}(\delta, p)}{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}$ $\frac{\langle \sigma_{\text{abs}}^0(x), p \rangle \xrightarrow{a} \langle x', p \rangle}{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}$ $\frac{\langle \sigma_{\text{abs}}^r(x), p+r \rangle \xrightarrow{a} \langle \sigma_{\text{abs}}^r(x'), p+r \rangle}{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r \langle x, p+r \rangle}{\langle \sigma_{\text{abs}}^q(x), p+q \rangle \vdash^r \langle \sigma_{\text{abs}}^q(x), p+q+r \rangle}$ $\frac{q > p}{\langle \sigma_{\text{abs}}^{q+r}(x), p \rangle \vdash^r \langle \sigma_{\text{abs}}^{q+r}(x), p+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}{\langle x+y, p \rangle \xrightarrow{a} \langle x', p \rangle, \langle y+x, p \rangle \xrightarrow{a} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r \langle x, p+r \rangle}{\langle x+y, p \rangle \vdash^r \langle x+y, p+r \rangle, \langle y+x, p \rangle \vdash^r \langle y+x, p+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle}{\langle x \cdot y, p \rangle \xrightarrow{a} \langle x' \cdot y, p \rangle}$ $\frac{\langle x, p \rangle \vdash^r \langle x, p+r \rangle}{\langle x \cdot y, p \rangle \vdash^r \langle x \cdot y, p+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle, q > p}{\langle v_{\text{abs}}^q(x), p \rangle \xrightarrow{a} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r \langle x, p+r \rangle, q > p+r}{\langle v_{\text{abs}}^q(x), p \rangle \vdash^r \langle v_{\text{abs}}^q(x), p+r \rangle}$ $\frac{q \leq p}{\text{ID}(v_{\text{abs}}^q(x), p)}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle, q \leq p}{\langle \bar{v}_{\text{abs}}^q(x), p \rangle \xrightarrow{a} \langle x', p \rangle}$ $\frac{\langle x, p \rangle \vdash^r \langle x, p+r \rangle, q \leq p+r}{\langle \bar{v}_{\text{abs}}^q(x), p \rangle \vdash^r \langle \bar{v}_{\text{abs}}^q(x), p+r \rangle}$ $\frac{q > p}{\langle \bar{v}_{\text{abs}}^{q+r}(x), p \rangle \vdash^r \langle \bar{v}_{\text{abs}}^{q+r}(x), p+r \rangle}$ | $\frac{\langle \tilde{a}, 0 \rangle \xrightarrow{a} \langle \sqrt{\cdot}, 0 \rangle}{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\langle \sigma_{\text{abs}}^0(x), p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\langle \sigma_{\text{abs}}^r(x), p+r \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p+r \rangle}{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\text{ID}(x, p)}{\text{ID}(\sigma_{\text{abs}}^q(x), p+q)}$ $\frac{\neg \text{ID}(x, 0)}{\langle \sigma_{\text{abs}}^{q+r}(x), q \rangle \vdash^r \langle \sigma_{\text{abs}}^{q+r}(x), q+r \rangle}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}{\langle x+y, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, \langle y+x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\text{ID}(x, p), \text{ID}(y, p)}{\text{ID}(x+y, p)}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}{\langle x \cdot y, p \rangle \xrightarrow{a} \langle y, p \rangle}$ $\frac{\text{ID}(x, p)}{\text{ID}(x \cdot y, p)}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, q > p}{\langle v_{\text{abs}}^q(x), p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\text{ID}(x, p), q > p}{\text{ID}(v_{\text{abs}}^q(x), p)}$ $\frac{\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, q \leq p}{\langle \bar{v}_{\text{abs}}^q(x), p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle}$ $\frac{\text{ID}(x, p), q \leq p}{\text{ID}(\bar{v}_{\text{abs}}^q(x), p)}$ $\frac{\neg \text{ID}(x, q+r)}{\langle \bar{v}_{\text{abs}}^{q+r}(x), q \rangle \vdash^r \langle \bar{v}_{\text{abs}}^{q+r}(x), q+r \rangle}$ |
|---|--|

Table 20: Rules for operational semantics of BPA^{sat} ($a \in \mathbf{A}$, $r > 0$, $p, q \geq 0$)

| | |
|---|--|
| $\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle, \neg \text{ID}(y, p)$ | |
| $\langle x \parallel y, p \rangle \xrightarrow{a} \langle x' \parallel y, p \rangle, \langle y \parallel x, p \rangle \xrightarrow{a} \langle y \parallel x', p \rangle, \langle x \ll y, p \rangle \xrightarrow{a} \langle x' \parallel y, p \rangle$ | |
| $\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, \neg \text{ID}(y, p)$ | |
| $\langle x \parallel y, p \rangle \xrightarrow{a} \langle y, p \rangle, \langle y \parallel x, p \rangle \xrightarrow{a} \langle y, p \rangle, \langle x \ll y, p \rangle \xrightarrow{a} \langle y, p \rangle$ | |
| $\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle, \langle y, p \rangle \xrightarrow{b} \langle y', p \rangle, \gamma(a, b) = c$ | |
| $\langle x \parallel y, p \rangle \xrightarrow{c} \langle x' \parallel y', p \rangle, \langle x \mid y, p \rangle \xrightarrow{c} \langle x' \parallel y', p \rangle$ | |
| $\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle, \langle y, p \rangle \xrightarrow{b} \langle \sqrt{\cdot}, p \rangle, \gamma(a, b) = c$ | |
| $\langle x \parallel y, p \rangle \xrightarrow{c} \langle x', p \rangle, \langle y \parallel x, p \rangle \xrightarrow{c} \langle x', p \rangle,$ $\langle x \mid y, p \rangle \xrightarrow{c} \langle x', p \rangle, \langle y \mid x, p \rangle \xrightarrow{c} \langle x', p \rangle$ | |
| $\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, \langle y, p \rangle \xrightarrow{b} \langle \sqrt{\cdot}, p \rangle, \gamma(a, b) = c$ | |
| $\langle x \parallel y, p \rangle \xrightarrow{c} \langle \sqrt{\cdot}, p \rangle, \langle x \mid y, p \rangle \xrightarrow{c} \langle \sqrt{\cdot}, p \rangle$ | |
| $\langle x, p \rangle \vdash^r \langle x, p + r \rangle, \langle y, p \rangle \vdash^r \langle y, p + r \rangle$ | |
| $\langle x \parallel y, p \rangle \vdash^r \langle x \parallel y, p + r \rangle, \langle x \ll y, p \rangle \vdash^r \langle x \ll y, p + r \rangle,$ $\langle x \mid y, p \rangle \vdash^r \langle x \mid y, p + r \rangle$ | |
| $\text{ID}(x, p)$ | |
| $\text{ID}(x \parallel y, p), \text{ID}(y \parallel x, p), \text{ID}(x \ll y, p),$ $\text{ID}(y \ll x, p), \text{ID}(x \mid y, p), \text{ID}(y \mid x, p)$ | |
| $\langle x, p \rangle \xrightarrow{a} \langle x', p \rangle, a \notin H$ | $\langle x, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, a \notin H$ |
| $\langle \partial_H(x), p \rangle \xrightarrow{a} \langle \partial_H(x'), p \rangle$ | $\langle \partial_H(x), p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle$ |
| $\langle x, p \rangle \vdash^r \langle x, p + r \rangle$ | $\text{ID}(x, p)$ |
| $\langle \partial_H(x), p \rangle \vdash^r \langle \partial_H(x), p + r \rangle$ | $\text{ID}(\partial_H(x), p)$ |
| $\langle x, 0 \rangle \xrightarrow{a} \langle x', 0 \rangle$ | $\langle x, 0 \rangle \xrightarrow{a} \langle \sqrt{\cdot}, 0 \rangle$ |
| $\langle \nu_{\text{abs}}(x), 0 \rangle \xrightarrow{a} \langle x', 0 \rangle$ | $\langle \nu_{\text{abs}}(x), 0 \rangle \xrightarrow{a} \langle \sqrt{\cdot}, 0 \rangle$ |
| $\text{ID}(x, 0)$ | $\text{ID}(x, 0)$ |
| $\text{ID}(\nu_{\text{abs}}(x), 0)$ | $\text{ID}(\nu_{\text{abs}}(x), r)$ |

Table 21: Additional rules for ACP^{sat} ($a, b, c \in \mathbf{A}, r > 0, p \geq 0$)

| | |
|--|---|
| $\langle P[q/v], p \rangle \xrightarrow{a} \langle P', p \rangle, q \in V$ | $\langle P[q/v], p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle, q \in V$ |
| $\langle \int_{v \in V} P, p \rangle \xrightarrow{a} \langle P', p \rangle$ | $\langle \int_{v \in V} P, p \rangle \xrightarrow{a} \langle \sqrt{\cdot}, p \rangle$ |
| $\langle P[q/v], p \rangle \vdash^r \langle P[q/v], p + r \rangle, q \in V$ | $\text{ID}(P[q/v], p)$ for all $q \in V$ |
| $\langle \int_{v \in V} P, p \rangle \vdash^r \langle \int_{v \in V} P, p + r \rangle$ | $\text{ID}(\int_{v \in V} P, p)$ |

Table 22: Rules for integration ($a \in \mathbf{A}, r > 0, p, q \geq 0$)

| | |
|---|---|
| $\dot{\delta} = \lambda t . \dot{\delta}$ | $f \parallel g = \lambda t . (f(t) \parallel g(t))$ |
| $\tilde{a} = \lambda t . \overline{v}_{\text{abs}}^t(\tilde{a})$ | $f \ll g = \lambda t . (f(t) \ll g(t))$ |
| $\sigma_{\text{abs}}^p(f) = \lambda t . \overline{v}_{\text{abs}}^t(\sigma_{\text{abs}}^p(f(0)))$ | $f \mid g = \lambda t . (f(t) \mid g(t))$ |
| $f + g = \lambda t . (f(t) + g(t))$ | $\partial_H(f) = \lambda t . \partial_H(f(t))$ |
| $f \cdot g = \lambda t . (f(t) * g)$ | $\nu_{\text{abs}}(f) = \lambda t . \overline{v}_{\text{abs}}^t(\nu_{\text{abs}}(f(t)))$ |
| $v_{\text{abs}}^p(f) = \lambda t . \overline{v}_{\text{abs}}^t(v_{\text{abs}}^p(f(t)))$ | $\int_{v \in V}(f) = \lambda t . \int_{v \in V}(f(t))$ |
| $\overline{v}_{\text{abs}}^p(f) = f(p)$ | $\sqrt{s}^* \varphi = \lambda t . \overline{v}_{\text{abs}}^t(\varphi(t))$ |

Table 23: Definition of operators on RTTS^* ($a \in \mathbf{A}_\delta$, $p \in \mathbb{R}_{\geq 0}$, $\varphi : \mathbb{R}_{\geq 0} \rightarrow \text{RTTS}^*(\mathbf{A})$)

B ACP^{dat} and initial abstraction

Axiom systems The axiom system of BPA^{dat} consists of the equations given in Table 24. The axiom system of ACP^{dat} consists of the equations given in Tables 24 and 25. The axioms for discrete initial abstraction are given in Table 26.

| | | | |
|--|-------|---|-------|
| $x + y = y + x$ | A1 | | |
| $(x + y) + z = x + (y + z)$ | A2 | | |
| $x + x = x$ | A3 | $v_{\text{abs}}^n(\dot{\delta}) = \dot{\delta}$ | DAT00 |
| $(x + y) \cdot z = (x \cdot z) + (y \cdot z)$ | A4 | $v_{\text{abs}}^0(x) = \dot{\delta}$ | DAT01 |
| $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ | A5 | $v_{\text{abs}}^{n+1}(\underline{a}) = \underline{a}$ | DAT02 |
| $x + \dot{\delta} = x$ | A6ID | $v_{\text{abs}}^{m+n}(\sigma_{\text{abs}}^n(x)) = \sigma_{\text{abs}}^n(v_{\text{abs}}^m(x))$ | DAT03 |
| $\dot{\delta} \cdot x = \dot{\delta}$ | A7ID | $v_{\text{abs}}^n(x + y) = v_{\text{abs}}^n(x) + v_{\text{abs}}^n(y)$ | DAT04 |
| | | $v_{\text{abs}}^n(x \cdot y) = v_{\text{abs}}^n(x) \cdot y$ | DAT05 |
| $\sigma_{\text{abs}}^0(x) = \bar{v}_{\text{abs}}^0(x)$ | DAT1 | | |
| $\sigma_{\text{abs}}^m(\sigma_{\text{abs}}^n(x)) = \sigma_{\text{abs}}^{m+n}(x)$ | DAT2 | $\bar{v}_{\text{abs}}^0(\dot{\delta}) = \dot{\delta}$ | DAI0a |
| $\sigma_{\text{abs}}^n(x) + \sigma_{\text{abs}}^n(y) = \sigma_{\text{abs}}^n(x + y)$ | DAT3 | $\bar{v}_{\text{abs}}^{n+1}(\dot{\delta}) = \sigma_{\text{abs}}^{n+1}(\dot{\delta})$ | DAI0b |
| $\sigma_{\text{abs}}^n(x) \cdot v_{\text{abs}}^n(y) = \sigma_{\text{abs}}^n(x \cdot \dot{\delta})$ | DAT4 | $\bar{v}_{\text{abs}}^0(\underline{a}) = \underline{a}$ | DAI1 |
| $\sigma_{\text{abs}}^n(x) \cdot (v_{\text{abs}}^n(y) + \sigma_{\text{abs}}^n(z)) =$ | | $\bar{v}_{\text{abs}}^{n+1}(\underline{a}) = \sigma_{\text{abs}}^{n+1}(\dot{\delta})$ | DAI2 |
| $\sigma_{\text{abs}}^n(x \cdot \bar{v}_{\text{abs}}^0(z))$ | DAT5 | $\bar{v}_{\text{abs}}^{m+n}(\sigma_{\text{abs}}^n(x)) = \sigma_{\text{abs}}^n(\bar{v}_{\text{abs}}^m(\bar{v}_{\text{abs}}^0(x)))$ | DAI3 |
| $\sigma_{\text{abs}}^n(\dot{\delta}) \cdot x = \sigma_{\text{abs}}^n(\dot{\delta})$ | DAT6 | $\bar{v}_{\text{abs}}^n(x + y) = \bar{v}_{\text{abs}}^n(x) + \bar{v}_{\text{abs}}^n(y)$ | DAI4 |
| $\sigma_{\text{abs}}^1(\dot{\delta}) = \underline{\delta}$ | DAT7 | $\bar{v}_{\text{abs}}^n(x \cdot y) = \bar{v}_{\text{abs}}^n(x) \cdot y$ | DAI5 |
| $\underline{a} + \underline{\delta} = \underline{a}$ | A6DAa | | |

Table 24: Axioms for BPA^{dat} ($a \in \mathbf{A}_\delta$)

| | | | |
|--|-------|---|-------|
| $\underline{a} \mid \underline{b} = \underline{c}$ if $\gamma(a, b) = c$ | CF1DA | $\underline{a} \cdot x \mid \underline{b} \cdot y = (\underline{a} \mid \underline{b}) \cdot (x \parallel y)$ | CM7DA |
| $\underline{a} \mid \underline{b} = \underline{\delta}$ if $\gamma(a, b)$ undefined | CF2DA | $(v_{\text{abs}}^1(x) + \underline{\delta}) \mid \sigma_{\text{abs}}^{n+1}(y) = \underline{\delta}$ | DACM3 |
| | | $\sigma_{\text{abs}}^{n+1}(x) \mid (v_{\text{abs}}^1(y) + \underline{\delta}) = \underline{\delta}$ | DACM4 |
| $x \parallel y = (x \parallel y + y \parallel x) + x \mid y$ | CM1 | $\sigma_{\text{abs}}^n(x) \mid \sigma_{\text{abs}}^n(y) = \sigma_{\text{abs}}^n(x \mid y)$ | DACM5 |
| $\dot{\delta} \parallel x = \dot{\delta}$ | CMID1 | $(x + y) \mid z = x \mid z + y \mid z$ | CM8 |
| $x \parallel \dot{\delta} = \dot{\delta}$ | CMID2 | $x \mid (y + z) = x \mid y + x \mid z$ | CM9 |
| $\underline{a} \parallel (x + \underline{\delta}) = \underline{a} \cdot (x + \underline{\delta})$ | CM2DA | | |
| $\underline{a} \cdot x \parallel (y + \underline{\delta}) = \underline{a} \cdot (x \parallel (y + \underline{\delta}))$ | CM3DA | $\partial_H(\dot{\delta}) = \dot{\delta}$ | D0 |
| $\sigma_{\text{abs}}^n(x) \parallel (v_{\text{abs}}^n(y) + \sigma_{\text{abs}}^n(z)) = \sigma_{\text{abs}}^n(x \parallel z)$ | DACM2 | $\partial_H(\underline{a}) = \underline{a}$ if $a \notin H$ | D1DA |
| $(x + y) \parallel z = x \parallel z + y \parallel z$ | CM4 | $\partial_H(\underline{a}) = \underline{\delta}$ if $a \in H$ | D2DA |
| $\dot{\delta} \mid x = \dot{\delta}$ | CMID3 | $\partial_H(\sigma_{\text{abs}}^n(x)) = \sigma_{\text{abs}}^n(\partial_H(x))$ | DAD |
| $x \mid \dot{\delta} = \dot{\delta}$ | CMID4 | $\partial_H(x + y) = \partial_H(x) + \partial_H(y)$ | D3 |
| $\underline{a} \cdot x \mid \underline{b} = (\underline{a} \mid \underline{b}) \cdot x$ | CM5DA | $\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$ | D4 |
| $\underline{a} \mid \underline{b} \cdot x = (\underline{a} \mid \underline{b}) \cdot x$ | CM6DA | | |

Table 25: Additional axioms for ACP^{dat} ($a, b \in \mathbf{A}_\delta, c \in \mathbf{A}$)

| | | | |
|---|------|---|-------|
| $\sqrt{d} j . G = \sqrt{d} i . G[i/j]$ | DIA1 | $(\sqrt{d} i . F) + G = \sqrt{d} i . (F + \bar{v}_{\text{abs}}^i(G))$ | DIA8 |
| $\bar{v}_{\text{abs}}^n(\sqrt{d} i . F) = \bar{v}_{\text{abs}}^n(F[n/i])$ | DIA2 | $(\sqrt{d} i . F) \cdot G = \sqrt{d} i . (F \cdot G)$ | DIA9 |
| $\sqrt{d} i . (\sqrt{d} j . F) = \sqrt{d} i . F[i/j]$ | DIA3 | $v_{\text{abs}}^n(\sqrt{d} i . F) = \sqrt{d} i . v_{\text{abs}}^n(F)$ if $n \neq i$ | DIA10 |
| $G = \sqrt{d} i . G$ | DIA4 | $(\sqrt{d} i . F) \parallel G = \sqrt{d} i . (F \parallel \bar{v}_{\text{abs}}^i(G))$ | DIA11 |
| $(\forall n \in \mathbb{N} \bullet \bar{v}_{\text{abs}}^n(x) = \bar{v}_{\text{abs}}^n(y)) \Rightarrow$ | | $G \parallel (\sqrt{d} i . F) = \sqrt{d} i . (\bar{v}_{\text{abs}}^i(G) \parallel F)$ | DIA12 |
| $x = y$ | DIA5 | $(\sqrt{d} i . F) G = \sqrt{d} i . (F \bar{v}_{\text{abs}}^i(G))$ | DIA13 |
| $\sigma_{\text{abs}}^n(\underline{a}) \cdot x = \sigma_{\text{abs}}^n(\underline{a}) \cdot \bar{v}_{\text{abs}}^n(x)$ | DIA6 | $G (\sqrt{d} i . F) = \sqrt{d} i . (\bar{v}_{\text{abs}}^i(G) F)$ | DIA14 |
| $\sigma_{\text{abs}}^n(\sqrt{d} i . F) = \sigma_{\text{abs}}^n(F[0/i])$ | DIA7 | $\partial_H(\sqrt{d} i . F) = \sqrt{d} i . \partial_H(F)$ | DIA15 |

Table 26: Axioms for discrete initial abstraction (i not free in G)

C Outline of proofs

Proof of Lemma 3.

1. It is easy to prove by induction on the structure of b that $b = \sqrt{s}v \cdot \bar{v}_{\text{abs}}^{[v]}(b)$.

2. Lemma 3.2 is Lemma 7 from [6] for the case with conditionals. Therefore, it suffices to extend the proof by induction on the structure of t with the case that t is of the form $b :: \rightarrow t'$:

$$\begin{aligned}
b :: \rightarrow t' &\stackrel{\text{IH}}{=} b :: \rightarrow \sqrt{s}v \cdot \bar{v}_{\text{abs}}^{[v]}(t') \stackrel{\text{SASGC10}}{=} \sqrt{s}v \cdot (\bar{v}_{\text{abs}}^v(b) :: \rightarrow \bar{v}_{\text{abs}}^{[v]}(t')) \stackrel{\text{A6ID, Case } t \equiv \dot{\delta}, \text{DISTR}^+}{=} \\
&\sqrt{s}v \cdot ((\bar{v}_{\text{abs}}^v(b) :: \rightarrow \bar{v}_{\text{abs}}^{[v]}(t')) + \bar{v}_{\text{abs}}^{[v]}(\dot{\delta})) \stackrel{\text{Lemma 3.1, CSIA2, S118}}{=} \\
&\sqrt{s}v \cdot ((\bar{v}_{\text{abs}}^{[v]}(b) :: \rightarrow \bar{v}_{\text{abs}}^{[v]}(t')) + \bar{v}_{\text{abs}}^{[v]}(\dot{\delta})) \stackrel{\text{SAI0, SAT1}}{=} \\
&\sqrt{s}v \cdot ((\bar{v}_{\text{abs}}^{[v]}(b) :: \rightarrow \bar{v}_{\text{abs}}^{[v]}(t')) + \sigma_{\text{abs}}^{[v]}(\dot{\delta})) \stackrel{\text{SASGC1}}{=} \sqrt{s}v \cdot \bar{v}_{\text{abs}}^{[v]}(b :: \rightarrow t') \quad \square
\end{aligned}$$

Proof of Lemma 4. Lemma 4 is Lemma 9 from [6] adapted to the case with conditionals. The condition $\bar{v}_{\text{abs}}^p(t) \neq \sigma_{\text{abs}}^p(\dot{\delta})$ needed in the case with conditionals implies the condition $t \neq \dot{\delta}$ used in [6]. There, observing that the lemma would follow immediately, we only proved by induction on the structure of t that there exists a t' such that: (1) $\bar{v}_{\text{abs}}^p(t) = \sigma_{\text{abs}}^p(t')$ and (2) if $p \in [0, 1)$ and $\bar{v}_{\text{abs}}^p(t) \neq \sigma_{\text{abs}}^p(\dot{\delta})$, $t' = t + \sigma_{\text{abs}}^{1-p}(\dot{\delta})$. Here, it suffices to extend that proof with the case that t is of the form $b :: \rightarrow t'$:

$$\begin{aligned}
1. \quad &\bar{v}_{\text{abs}}^p(b :: \rightarrow t') \stackrel{\text{SASGC1}}{=} (\bar{v}_{\text{abs}}^p(b) :: \rightarrow \bar{v}_{\text{abs}}^p(t')) + \sigma_{\text{abs}}^p(\dot{\delta}) \stackrel{\text{IH}}{=} \\
&(\bar{v}_{\text{abs}}^p(b) :: \rightarrow \sigma_{\text{abs}}^p(t'')) + \sigma_{\text{abs}}^p(\dot{\delta}) \stackrel{\text{SASGC3}}{=} \sqrt{s}v \cdot \sigma_{\text{abs}}^p(\bar{v}_{\text{abs}}^v(\bar{v}_{\text{abs}}^p(b)) :: \rightarrow t'') \stackrel{\text{S118, SIA4}}{=} \\
&\sigma_{\text{abs}}^p(\bar{v}_{\text{abs}}^p(b) :: \rightarrow t'') \\
2. \quad &\bar{v}_{\text{abs}}^p(b :: \rightarrow t') \neq \sigma_{\text{abs}}^p(\dot{\delta}) \stackrel{\text{SGC1, SGC2ID, SASGC1}}{\Rightarrow} \bar{v}_{\text{abs}}^p(b) = t \text{ and } \bar{v}_{\text{abs}}^p(t') \neq \sigma_{\text{abs}}^p(\dot{\delta}) \\
&\text{By the induction hypothesis,} \\
&\bar{v}_{\text{abs}}^p(b) :: \rightarrow t'' = \bar{v}_{\text{abs}}^p(b) :: \rightarrow (t'' + \sigma_{\text{abs}}^{1-p}(\dot{\delta})) \stackrel{\text{SGC1}}{=} (\bar{v}_{\text{abs}}^p(b) :: \rightarrow t'') + \sigma_{\text{abs}}^{1-p}(\dot{\delta}) \quad \square
\end{aligned}$$

Proof of Lemma 5. Lemma 5 is Lemma 10 from [6] adapted to the case with conditionals. The form $\int_{v \in [0,1]} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t') + \tilde{\delta})$ realizable in the case with conditionals generalizes the form $\int_{v \in [0,1]} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t') + \tilde{\delta})$ obtained in [6]. Hence, it suffices to extend the proof by induction on the structure of t with the case that t is of the form $b :: \rightarrow t'$:

$$\begin{aligned}
v_{\text{abs}}^1(b :: \rightarrow t' + \underline{\delta}) &\stackrel{\text{SATO4, SASGC4}}{=} (b :: \rightarrow v_{\text{abs}}^1(t')) + v_{\text{abs}}^1(\underline{\delta}) \stackrel{\text{SGC1,6, BOOL4,6}}{=} \\
(b :: \rightarrow v_{\text{abs}}^1(t')) + (b :: \rightarrow v_{\text{abs}}^1(\underline{\delta})) + v_{\text{abs}}^1(\underline{\delta}) &\stackrel{\text{SATO0,3,6}}{=} \\
(b :: \rightarrow v_{\text{abs}}^1(t')) + (b :: \rightarrow v_{\text{abs}}^1(\underline{\delta})) + \underline{\delta} &\stackrel{\text{SGC4, SATO4}}{=} (b :: \rightarrow v_{\text{abs}}^1(t' + \underline{\delta})) + \underline{\delta} \stackrel{\text{IH}}{=} \\
(b :: \rightarrow \int_{v \in [0,1]} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t'') + \tilde{\delta})) + \underline{\delta} &\stackrel{\text{SAT3, INT10}}{=} \\
(b :: \rightarrow (\int_{v \in [0,1]} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t'')) + \underline{\delta})) + \underline{\delta} &\stackrel{\text{SGC1,6, BOOL4,6}}{=} \\
(b :: \rightarrow \int_{v \in [0,1]} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t''))) + \underline{\delta} &\stackrel{\text{SASGC9}}{=} \int_{v \in [0,1]} (b :: \rightarrow \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t''))) + \underline{\delta} \stackrel{\text{INT10}}{=} \\
\int_{v \in [0,1]} ((b :: \rightarrow \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t''))) + \sigma_{\text{abs}}^v(\tilde{\delta})) &\stackrel{\text{A6ID, SAT3}}{=} \\
\int_{v \in [0,1]} ((b :: \rightarrow \sigma_{\text{abs}}^v(\nu_{\text{abs}}(t''))) + \sigma_{\text{abs}}^v(\dot{\delta}) + \sigma_{\text{abs}}^v(\tilde{\delta})) &\stackrel{\text{SASGC3, SIA4}}{=} \\
\int_{v \in [0,1]} ((\sqrt{s}w \cdot \sigma_{\text{abs}}^v(\bar{v}_{\text{abs}}^w(b) :: \rightarrow \nu_{\text{abs}}(t''))) + \sqrt{s}w \cdot \sigma_{\text{abs}}^v(\tilde{\delta})) &\stackrel{\text{DISTR}^+, \text{SAT3}}{=} \\
\int_{v \in [0,1]} \sqrt{s}w \cdot \sigma_{\text{abs}}^v((\bar{v}_{\text{abs}}^w(b) :: \rightarrow \nu_{\text{abs}}(t''))) + \tilde{\delta} &\stackrel{\text{SASGC8, SIA17}}{=} \\
\sqrt{s}w \cdot \int_{v \in [0,1]} \sigma_{\text{abs}}^v(\nu_{\text{abs}}(\bar{v}_{\text{abs}}^w(b) :: \rightarrow t'') + \tilde{\delta}) &\quad \square
\end{aligned}$$

Proof of Theorem 6. Theorem 6 is Theorem 12 from [6] adapted to the case with conditionals. In [6], it is shown that the axioms of $\text{ACP}^{\text{dat}}\checkmark$ are derivable for closed

terms from the axioms of $\text{ACP}^{\text{sat}}\text{IV}$ and the explicit definitions of the constants and operators \underline{a} , σ_{abs} , v_{abs} , $\overline{v}_{\text{abs}}$ and \sqrt{d} (for processes) in Table 12. In [6], use is made of two lemmas that do not go through for the extension with conditionals, viz. Lemmas 9 and 10 from that paper. In the case with conditionals, Lemmas 4 and 5 from this paper have to be used instead. Fortunately, this requires only minor changes to the proofs for four axioms, viz. CM2DA, CM3DA, DACM3 and DACM4.

What remains to be shown is that the additional axioms for conditionals are derivable for closed terms. This is nontrivial for the following axioms: CDAI3-CDAI7, CDIA1-CDIA8, DASGC2, DASGC3, DASGC8 and DASGC9. However, the proofs for most of these axioms are either similar to proofs for axioms of $\text{ACP}^{\text{dat}}\text{V}$ (CDIA1-CDIA8, DASGC8 and DASGC9) or simpler than most of those proofs (CDAI3-CDAI7 and DASGC3). Therefore, we only give an idea of the proofs.

The proofs for axioms CDAI3-CDAI7 require little effort. They involve short calculations using axioms BOOL1-BOOL7 and CSAI1-CSAI10.

The proofs for axioms CDIA1-CDIA5 are analogous to the proofs for DIA1-DIA5 in [6] – axioms CSIA1-CSIA5 are used instead of axioms SIA1-SIA5.

The proof for axiom CDIA6 is similar to the proof for DIA10 in [6] – axiom CSIA6 is used instead of axiom SIA10.

The proof for axioms CDIA7 and CDIA8 are similar to the proof for DIA8 in [6] – axioms CSIA7 and CSIA8 are used instead of axiom SIA8. Distributivity of initial abstraction over \wedge and \vee is needed, but that can be derived as in the case of $+$.

The proof for axiom DASGC2 goes as follows. First of all, prove (1) $\text{sl}(n+1) ::\rightarrow x = \int_{v \in [n, n+1)} (\text{pt}(v) ::\rightarrow x)$, mainly by short calculations using axioms BOOL1-BOOL7 and CSAI1-CSAI10, and (2) $x = x + (b ::\rightarrow x)$, by application of axioms SGC1, SGC6 and BOOL4. Then, having proven equations (1) and (2), the proof for axiom DASGC2 involves mainly application of axiom SASGC2, these equations and the following immediate consequence of Lemma 3.2 and axiom SIA2: $\overline{v}_{\text{abs}}^p(\overline{v}_{\text{abs}}^p(t)) = \overline{v}_{\text{abs}}^p(\overline{v}_{\text{abs}}^{\lfloor p \rfloor}(t))$.

The proof for axiom DASGC3 is very easy. It consists of applying axiom SASGC3 and the following immediate consequence of Lemma 3.1 and axioms CSIA2 and SI18: $\overline{v}_{\text{abs}}^p(b) = \overline{v}_{\text{abs}}^{\lfloor p \rfloor}(b)$.

The proofs for axioms DASGC8 and DASGC9 are again similar to the proof for DIA8 – axioms SASGC10 and SASGC11 are used instead of axiom SIA8. Distributivity of initial abstraction over $::\rightarrow$ is needed, but that can be derived as in the case of $+$. \square