An Effective Axiomatization for Real Time ACP*

Wan Fokkink† & Steven Klusener
Department of Computer Science
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
wan@cwi.nl stevenk@cwi.nl

Baeten and Bergstra added real time to ACP, and introduced the notion of integration, which expresses the possibility of an action happening within a time interval. In order to axiomatize this feature, they needed an 'uncountable' axiom. This paper deals with prefix integration, and integration is parametrized by conditions, which are inequalities between linear expressions of variables. We present an axiomatization for process terms, and propose a strategy to decide bisimulation equivalence between process terms, by means of this axiomatization.

1 Introduction

In recent years, many process algebras have been extended with some notion of time. This paper is based on the approach of Baeten and Bergstra (1991), who extended ACP with real time. Their algebra concerns (closed) process terms, constructed from timed actions $a(t)$, which denote the process that executes action a at time t. This results in identities that do not hold in untimed ACP, such as

$$a(2) \cdot (b(1) + c(3)) = a(2) \cdot c(3).$$

After the execution of a, time has passed 2, so in the remaining subterm $b(1) + c(3)$ the first alternative is lost.

In Baeten and Bergstra (1991), the notion of integration was introduced, which expresses the possibility that an action occurs somewhere within a time interval. The construct $\int_{v\in V} p$ executes the process p, where the behaviour of p may depend on the value of v in the time interval V. In Baeten and Bergstra (1991), an axiomatization was proposed for processes that are time-closed, which means that if a process depends on a variable v, then it is guarded by some integral sign $\int_{v\in V}$. One of their axioms considerably hampers reasoning within the algebra, since in order to apply it one needs to check infinitely many equalities. Namely, this axiom says that two processes $\int_{v\in V} p$ and $\int_{v\in V} q$ are equal if p and q are equal for all possible values for v in the interval V.

*This paper is a combination of revised versions of Klusener (1991a) and Fokkink (1991).

†Current affiliation: Department of Philosophy, Utrecht University, Utrecht, The Netherlands, fokkink@phil.ruu.nl.

In this paper, we show how to get rid of this axiom. We restrict to prefix integration, and integration is parametrized by conditions, which consist of inequalities between linear expressions of variables. Furthermore, the notion of a conditional term is introduced, which is of the form $\phi :\rightarrow p$, where ϕ is a condition. The process $\phi :\rightarrow p$ executes p under the condition that ϕ is true. We present an axiomatization for time-closed conditional terms, which allows to mix conditions through terms by the following two equations:

$$\int_\phi a(v) \cdot p = \int_\phi a(v) \cdot (\phi :\rightarrow p),$$

$$\phi :\rightarrow \int_\psi a(v) \cdot p = \int_{\phi \wedge \psi} a(v) \cdot p \qquad \text{if } v \text{ does not occur in } \phi.$$

Moreover, we present axiomatizations for bounds and for conditions.

Conditional terms can be reduced to a normal form, using the axioms, such that if two time-closed processes are bisimilar, then they have the same normal form. Hence, the axiom system decides bisimulation equivalence between time-closed processes.

See Klusener (1991b, 1992) how to deal with abstraction in our setting, modulo timed branching bisimulation. A thorough treatment of real time ACP can be found in Klusener (1993). Section 5.2 provides a comparison of real time ACP and related timed process algebras, which incorporate some construct to express time dependencies.

2 The Syntax and Semantics

This section contains a description of the syntax and semantics for BPAδ with real time and prefix integration and conditions.

2.1 Bounds and conditions

In the sequel, we assume a countably infinite set of time variables *TVar*. Furthermore, we assume a set of time numbers *Time* which is a field. So *Time* is closed under the binary operations addition and multiplication, which are associative and commutative and satisfy the distributivity laws. Moreover, it contains distinct units 0 and 1 for addition and multiplication respectively, and for each time number $t \neq 0$, there are time numbers $-t$ and t^{-1} such that $t + (-t) = 0$ and $t \cdot t^{-1} = 1$. We assume a (reflexive and transitive) total ordering $\leq$ on *Time*, which is preserved under addition and multiplication with time numbers greater than 0.

The full domain of time numbers will only be used in the operational semantics of process terms. In order to build the syntax of process terms from a finite signature, this syntax uses a countable sub-field $Time_0$ of *Time*, which is defined as follows. Fix functions $f_i : Time^{m_i} \rightarrow Time$ for $i = 1, ..., k$, and let $Time_0$ be the field that is generated by $f_1, ..., f_k$, i.e. $Time_0$ is the smallest sub-field of *Time* such that if $r_1, ..., r_{m_i} \in Time_0$, then $f_i(r_1, ..., r_{m_i}) \in Time_0$. (In the examples, $Time_0$ will consist of the rational numbers.) We assume that for each pair of time numbers r_0 and r_1 in $Time_0$ it can be decided whether or not $r_0 \leq r_1$ holds.

Let $r \in Time_0$ and $v \in TVar$. The set of *bounds*, with typical element b, is defined by

$$b \ ::= \ r \ | \ v \ | \ b + b \ | \ r \cdot b.$$

The set of time variables that occur in a bound b is denoted by $var(b)$. The expression $b_0 - b_1$ abbreviates $b_0 + (-1) \cdot b_1$.

A *condition* is constructed from conjunctions and negations of inequalities between bounds. So the set of conditions, with typical element ϕ, is defined by

$$\phi \ ::= \ b_0 \leq b_1 \ \mid \ \phi \wedge \phi \ \mid \ \neg\phi.$$

The set of time variables that occur in a condition ϕ is denoted by $var(\phi)$. In the sequel, we use the abbreviations $\phi \vee \psi$ for $\neg(\neg\phi \wedge \neg\psi)$, and *tt* for $\neg(1 \leq 0)$, and *ff* for $1 \leq 0$, and $b_0 < b_1$ for $\neg(b_1 \leq b_0)$, and $b_0 = b_1$ for $b_0 \leq b_1 \wedge b_1 \leq b_0$, and $b_0 < v < b_1$ for $b_0 < v \wedge v < b_1$.

2.2 Process terms

Assume a countable alphabet A of atomic actions, together with a special constant δ, representing deadlock. In the sequel, a and α denote elements of A and $A \cup \{\delta\}$ respectively.

Integration enables to express that the behaviour of a process may depend on the value of a time variable. If a process p depends on the value of v between 1 and 2, then we write

$$\int_{1<v<2} p.$$

Here, integration is parametrized by conditions, and we deal with *prefix* integration $\int_\phi \alpha(v) \cdot p$. If $\alpha \neq \delta$, then this process can execute the action $\alpha(t)$ under the condition that $\phi[t/v]$ is true, after which the process results to $p[t/v]$.

In BPA$\delta\rho I$, process terms are constructed from prefix integration, the alternative composition $p + q$, and the time shift $b \gg p$, where b is a bound. (We do not incorporate sequential composition $p \cdot q$ from untimed BPA in BPA$\delta\rho I$, because this construct can be eliminated from the syntax by a straightforward set of axioms, see Klusener (1991a).) The time shift is an auxiliary operator that is needed in the operational semantics of integration; the process $b \gg p$ displays the behaviour of p after time b. Finally, we introduce the conditional construct $\phi :\rightarrow p$, where ϕ is a condition. This process displays the behaviour of p under the condition that ϕ is true. Thus, the set of process terms, with typical element p, is defined by

$$p \ ::= \ \int_\phi \alpha(v) \ \mid \ \int_\phi \alpha(v) \cdot p \ \mid \ p + p \ \mid b \gg p \ \mid \ \phi :\rightarrow p.$$

As binding convention, integration and time shift bind stronger than alternative composition. Often $\alpha(b)$ will abbreviate $\int_{v=b} \alpha(v)$, where $v \notin var(b)$.

2.3 Free variables and substitutions

In general, one cannot attach a transition system to a process term that contains time variables which are not bound by an integral sign. Therefore, we need the notion of a time-closed process. In the term $\int_\phi \alpha(v)$, occurrences of v in ϕ are bound, and in $\int_\phi \alpha(v) \cdot p$, occurrences of v in ϕ and in p are bound. A time variable in a process term is called free if it is not bound by any integral signs. The collection of free variables in a term p is denoted by $fvar(p)$. A term p is called *time-closed* if $fvar(p) = \emptyset$. As a model we will take the collection $\mathcal{T}^{cl}$ of time-closed process terms, modulo bisimulation (see Section 2.5).

A substitution is a mapping from time variables to bounds. For b a bound and σ a substitution, $\sigma(b)$ denotes the bound that results from substituting $\sigma(v)$ for each time variable v in b. Substitutions extend to conditions as expected. A substitution $\sigma : TVar \to Time_0$ is called a *valuation*. For a condition ϕ and a valuation σ, $\sigma(\phi)$ results to either true or false. In the sequel, $[\phi]$ denotes the collection of valuations for which $\sigma(\phi)$ is true.

For a process term p, $\sigma(p)$ is obtained by replacing free occurrences of time variables v in p by $\sigma(v)$. In this definition however, there is one serious complication. Namely, if a free occurrence of v in p has been replaced by $\sigma(v)$, then variables w that occur in $\sigma(v)$ may suddenly be bound in subterms $\int_\phi \alpha(w) \cdot q$ of p. A solution for this problem, which originates from the λ-calculus, is to allow unrestricted substitution by renaming bound variables. In the sequel, process terms are considered modulo α-conversion, and when a substitution is applied, bound variables are renamed. Stoughton (1988) presented a simple treatment of this technique.

2.4 Operational semantics

Table 1 contains an operational semantics for the collection $\mathcal{T}^{cl}$ of time-closed processes, in the style of Plotkin. In this operational semantics, we need a more general notion of bounds (and thus of conditions and process terms), which may contain time numbers from *Time* instead of $Time_0$. That is, the BNF grammar of a bound in this section is $b ::= t \mid v \mid b + b \mid r \cdot b$, where t is allowed to be in *Time*.

In most timed process algebras, the passing of time is expressed by idle transitions. For example, the process $a(1)$ can do an idle transition to time t for $t < 1$, meaning that the process has reached time t. Finally, at time 1, it executes action a. Such an operational semantics can be found for timed CCS (Wang (1990)), timed CSP (Schneider (1991)) and timed ACP (Baeten and Bergstra (1991)). We abstract from idle transitions, so here $a(1)$ only executes the a at time 1. A similar operational semantics can be found in Holmer, Larsen, and Wang (1991).

The timed deadlock $\delta(t)$ idles until time t. For example, the process $\int_{v<1} a(v) + \delta(3)$ either executes a before time 1, or idles until time 3. On the other hand, the process $\int_{v<1} a(v) + \delta(1)$ will always execute a before time 1. In order to distinguish processes that only differ in their deadlock behaviour, we introduce the delay predicate $U_t(p)$, which holds if p can idle beyond time t. (Moller and Tofts (1990) introduced a similar construct.) Processes that only differ in their deadlock behaviour have distinct delays. For example, $U_2(\int_{v<1} a(v) + \delta(3))$, but not $U_2(\int_{v<1} a(v) + \delta(1))$.

2.5 Bisimulation

Time-closed process terms are considered modulo *(strong) bisimulation*, which takes into account delays.

Table 1: Action rules for time-closed terms

$$\frac{\phi[t/v]}{\int_\phi a(v) \xrightarrow{a(t)} \surd \qquad \int_\phi a(v)\cdot x \xrightarrow{a(t)} t \gg x[t/v]}$$

$$\frac{x \xrightarrow{a(t)} \surd}{x+y \xrightarrow{a(t)} \surd \quad y+x \xrightarrow{a(t)} \surd} \qquad \frac{x \xrightarrow{a(t)} x'}{x+y \xrightarrow{a(t)} x' \quad y+x \xrightarrow{a(t)} x'}$$

$$\frac{b<t \quad x \xrightarrow{a(t)} \surd}{b \gg x \xrightarrow{a(t)} \surd} \qquad \frac{b<t \quad x \xrightarrow{a(t)} x'}{b \gg x \xrightarrow{a(t)} x'}$$

$$\frac{\phi \quad x \xrightarrow{a(t)} \surd}{\phi :\to x \xrightarrow{a(t)} \surd} \qquad \frac{\phi \quad x \xrightarrow{a(t)} x'}{\phi :\to x \xrightarrow{a(t)} x'}$$

$$\frac{\phi[s/v] \quad t<s}{U_t(\int_\phi \alpha(v)) \quad U_t(\int_\phi \alpha(v)\cdot x)} \qquad \frac{U_t(x)}{U_t(x+y) \quad U_t(y+x)}$$

$$\frac{U_t(x)}{U_t(b \gg x)} \qquad \frac{t<b}{U_t(b \gg x)} \qquad \frac{\phi \quad U_t(x)}{U_t(\phi :\to x)}$$

Definition 2.1 *Two time-closed process terms p_0, q_0 are* strongly bisimilar, *denoted by $p_0 \leftrightarrow q_0$, if there exists a symmetric binary* bisimulation *relation $\mathcal{B}$ on time-closed process terms such that*

- $p_0 \mathcal{B} q_0$,
- *if $p \xrightarrow{a(t)} p'$ and $p\mathcal{B}q$, then $q \xrightarrow{a(t)} q'$ for some process q' with $p'\mathcal{B}q'$,*
- *if $p \xrightarrow{a(t)} \surd$ and $p\mathcal{B}q$, then $q \xrightarrow{a(t)} \surd$,*
- *if $p\mathcal{B}q$ and $U_t(p)$, then $U_t(q)$.*

The rules in Table 1 can be fitted into the congruence format for action rules with types, data and variable binding of Bloom and Vaandrager (1994). The action rule for integration has to be adapted in order to fit it in the congruence format. Let *Bool* denote the type of Booleans and *Proc* the type of process terms. Integration is a function $\int(\lambda v.\phi, a, \lambda v.p)$, where $\int : (Time \to Bool) \times A \times (Time \to Proc) \longrightarrow Proc$. The action rule for integration takes the form

$$\frac{\texttt{pick}\ t : Time \quad \phi\, t}{\int(\phi, a, x) \xrightarrow{\langle a,t\rangle} t \gg (x\ t)}.$$

Strong bisimulation defined by transition rules within the format of Bloom and Vaandrager is always a congruence on the algebra of closed terms, which means that if $p \underline{\leftrightarrow} p'$ and $q \underline{\leftrightarrow} q'$, then $p + q \underline{\leftrightarrow} p' + q'$ and $\int_\phi \alpha(v) \cdot p \underline{\leftrightarrow} \int_\phi \alpha(v) \cdot p'$ and $b \gg p \underline{\leftrightarrow} b \gg p'$.

2.6 Axioms for bounds and conditions

Table 2 contains an axiomatization BA for bounds. It is assumed that BA incorporates the equalities $r_0 + r_1 = r_2$ and $r_0 \cdot r_1 = r_2$ between time numbers in $Time_0$. We consider bounds modulo AC, that is, modulo associativity and commutativity of the $+$. Using the axioms of BA, bounds can be reduced to a normal form $r_1 \cdot v_1 + ... + r_n \cdot v_n + r$, where $v_1, ..., v_n$ are distinct variables and $r_1, ..., r_n$ are unequal to zero. Using these normal forms, it is easy to deduce the following proposition.

Proposition 2.2 $b_0 = b_1 \Leftrightarrow \sigma(b_0)$ *and* $\sigma(b_1)$ *result to the same time number for each valuation* σ.

Table 2: Axioms BA for bounds

$b_0 + b_1 = b_1 + b_0$
$(b_0 + b_1) + b_2 = b_0 + (b_1 + b_2)$
$b + 0 = b$
$1 \cdot b = b$
$0 \cdot b = 0$
$r \cdot (b_0 + b_1) = r \cdot b_0 + r \cdot b_1$
$r_0 \cdot b + r_1 \cdot b = (r_0 + r_1) \cdot b$
$r_0 \cdot (r_1 \cdot b) = (r_0 \cdot r_1) \cdot b$

Table 3 contains an axiom system CA for conditions. The construct $\phi \Rightarrow \psi$, which is used in two of the axioms, abbreviates $\neg\phi \vee \psi$. The six Boolean axioms are complete for the algebra generated by *tt* and $\wedge$ and $\neg$ (see e.g. Koppelberg (1989)), and the four ordering axioms for bounds characterize a linear ordered field (see e.g. Chang and Keisler (1990)). It is assumed that CA incorporates the axiom system BA for bounds, that is, bounds in a condition may be manipulated by axioms in BA. In the sequel, $\phi = \psi$ denotes that this equality between conditions can be derived from CA.

The following lemma will be crucial in several constructions. Note that it would not hold if we had allowed bounds of the form v^2.

Lemma 2.3 (refinement lemma) *Fix a time variable* v. *Each condition* ϕ *is provably equal to a condition of the form* $\vee_i(\phi_i \wedge \phi'_i)$, *where*

- $var(\phi_i) \subseteq var(\phi)\backslash\{v\}$,
- ϕ'_i *is of the form* $v = b$ *or* $v < b$ *or* $b < v$ *or* $b < v < b'$, *with* $var(b+b') \subseteq var(\phi)\backslash\{v\}$.

Table 3: Axioms CA for conditions

$\phi \wedge \psi \ = \ \psi \wedge \phi$	$b_0 + b \leq b_1 + b \ = \ b_0 \leq b_1$
$(\phi_0 \wedge \phi_1) \wedge \phi_2 \ = \ \phi_0 \wedge (\phi_1 \wedge \phi_2)$	$r \cdot b_0 \leq r \cdot b_1 \ = \ b_0 \leq b_1 \quad \text{if } r > 0$
$\phi \wedge (\psi_0 \vee \psi_1) \ = \ (\phi \wedge \psi_0) \vee (\phi \wedge \psi_1)$	$b_0 \leq b \wedge b \leq b_1 \ \Rightarrow \ b_0 \leq b_1$
$\phi \ \Rightarrow \ \phi \vee \psi$	$b_0 \leq b_1 \vee b_1 \leq b_0 \ = \ tt$
$\phi \wedge \neg\phi = \mathit{ff}$	
$\neg\neg\phi \ = \ \phi$	

Moreover, for each $\phi_i \wedge \phi_i'$ were ϕ_i' is of the form $b < v < b'$ we can guarantee that $\phi_i \Rightarrow b < b'$.

The axiom system CA is sound and complete in the following sense.

Proposition 2.4 $\phi = \phi' \quad \Leftrightarrow \quad [\phi] = [\phi']$.

Soundness of CA can be verified by checking it for each axiom separately.

The proofs of the refinement lemma and of the completeness for CA are technical and do not contain any surprises. Hence, outlines of these proofs are provided in the appendix.

2.7 Axioms for process terms

From now on, occurrences of time numbers in bounds are restricted to $Time_0$ again. Table 4 contains an axiomatization for BPA$\delta\rho I$. The construct $P(v)$ represents expressions of the form $\alpha(v)$ and $\alpha(v)\cdot x$. The equational theory for BPA$\delta\rho I$ incorporates the axiomatizations CA of conditions and BA of bounds. That is, conditions in terms can be manipulated by means of axioms in CA and BA.

For each axiom $p = q$, and for each valuation σ, we have $\sigma(p) \underline{\leftrightarrow} \sigma(q)$. Hence, it is easy to see that the following proposition holds, which says that the axioms respect bisimulation equivalence between time-closed process terms. In the sequel, $p = q$ will mean that this equality between the terms p and q can be derived from the equational theory of BPA$\delta\rho I$.

Proposition 2.5 $\forall p, q \in \mathcal{T}^{cl}, \qquad p = q \implies p \underline{\leftrightarrow} q.$

The rest of this paper is devoted to proving that the equational theory for BPA$\delta\rho I$ is complete for time-closed process terms, i.e. if for $p, q \in \mathcal{T}^{cl}$ we have $p \underline{\leftrightarrow} q$, then $p = q$. We present an algorithm, based on the axioms, which decides bisimulation equivalence between time-closed process terms.

2.8 Basic terms

Definition 2.6 *A term is* basic *if it is in the class defined by*

$$p \ ::= \ \int_\phi \alpha(v) \ \mid \ \int_\phi \alpha(v) \cdot p \ \mid \ p + p,$$

Table 4: Axioms for BPA$\delta\rho I$

A1		$x + y$	$=$	$y + x$
A2		$(x + y) + z$	$=$	$x + (y + z)$
TA3		$\int_\phi P(v) + \int_\psi P(v)$	$=$	$\int_{\phi\vee\psi} P(v)$
TA4		$\int_\phi P(v) + \int_\phi \delta(v)$	$=$	$\int_\phi P(v)$
TA5		$\int_\phi \delta(v) \cdot x$	$=$	$\int_\phi \delta(v)$
TA6		$\int_{\mathit{ff}} P(v)$	$=$	$\int_{\mathit{ff}} \delta(v)$
TA7	$v \notin var(b)$	$\int_{v<b} \delta(v)$	$=$	$\delta(b)$
TS1		$\int_\phi \alpha(v) \cdot x$	$=$	$\int_\phi \alpha(v) \cdot (v \gg x)$
TS2	$v \notin var(b)$	$b \gg \int_\phi P(v)$	$=$	$\int_{\phi\wedge b<v} P(v) + \delta(b)$
S3		$b \gg (x + y)$	$=$	$b \gg x + b \gg y$
TC1		$\int_\phi \alpha(v) \cdot x$	$=$	$\int_\phi \alpha(v) \cdot (\phi :\rightarrow x)$
TC2	$v \notin var(\phi)$	$\phi :\rightarrow \int_\psi P(v)$	$=$	$\int_{\phi\wedge\psi} P(v)$
C3		$\phi :\rightarrow (x + y)$	$=$	$(\phi :\rightarrow x) + (\phi :\rightarrow y)$

and for each subterm $\int_\phi \alpha(v) \cdot p$ we have $\sigma(v \gg p) \leftrightarrow \sigma(p)$ for all valuations σ in $[\phi]$.

Proposition 2.7 *For each term p there is a basic term p' such that $p = p'$.*

Proof. First, we show how the axioms can be applied in order to get rid of time shifts and conditions. Let p_0 denote a term that does not contain time shifts nor conditions. Suppose that p is of the form $b \gg p_0$. We delete the time shift from p as follows, by induction on the *size* of p_0 (i.e. the number of function symbols in p_0). Ensure by means of α-conversion that $v \notin var(b)$.

$$\begin{aligned} b \gg \int_\phi P(v) &\overset{\text{TS2}}{=} \int_{\phi\wedge b<v} P(v) + \delta(b), \\ b \gg (q + q') &\overset{\text{S3}}{=} b \gg q + b \gg q'. \end{aligned}$$

Next, suppose that p is of the form $\phi :\rightarrow p_0$. We delete the condition from p as follows, by induction on the size of p_0. Ensure by means of α-conversion that $v \notin var(\phi)$.

$$\begin{aligned} \phi :\rightarrow \int_\psi P(v) &\overset{\text{TC2}}{=} \int_{\phi\wedge\psi} P(v), \\ \phi :\rightarrow (q + q') &\overset{\text{C3}}{=} (\phi :\rightarrow q) + (\phi :\rightarrow q'). \end{aligned}$$

If p contains several time shifts and conditions, then these operators can be deleted one by one, by considering subterms of p which contain only one time shift or one condition.

Finally, we show that p equals a basic term, by induction on size. A term $\int_\phi \alpha(v)$ is already basic, and if p and p' are basic, then $p + p'$ is basic too. Since we have deleted time shifts and conditions, this only leaves the case $\int_\phi \alpha(v) \cdot p$, where p is basic.

$$\begin{aligned} & \textstyle\int_\phi \alpha(v) \cdot \sum_i \int_{\psi_i} P_i(w) \\ \stackrel{\text{TS1}}{=} & \textstyle\int_\phi \alpha(v) \cdot v \gg \sum_i \int_{\psi_i} P_i(w) \\ \stackrel{\text{TS2,S3}}{=} & \textstyle\int_\phi \alpha(v) \cdot \sum_i (\int_{\psi_i \wedge v < w} P_i(w) + \delta(v)). \end{aligned}$$

This last term is basic. □

3 Unique Normal Forms

We shall describe a strategy which reduces each basic term p to a term which is called the *normal form* of p. All steps in the algorithm can be deduced from the axioms, so p is equal to its normal form. Next, we will show that if two time-closed normal forms are bisimilar, then they are equal modulo AC, i.e. modulo associativity and commutativity of the $+$. This will imply completeness of the axiom system. From now on, terms are considered modulo AC, and this equivalence is denoted by $=_{\mathrm{AC}}$.

In the following sections, we will present several equations between closed terms, that will be used in the construction of normal forms. These equations can all be deduced from the axioms. Finally, Section 3.6 provides a description of the construction of normal forms.

3.1 Some basic equations

The following equations for closed conditional terms can be deduced from the axioms.

Equation 3.1 $\phi :\to (\psi :\to p) \;=\; \phi \wedge \psi :\to p.$

Equation 3.2 $(\phi :\to p) + (\psi :\to p) \;=\; \phi \vee \psi :\to p.$

Equation 3.3 $p + \int_{\mathit{ff}} \delta(v) \;=\; p.$

Equation 3.4 $\mathit{tt} :\to p \;=\; p.$

Equation 3.5 $\mathit{ff} :\to p \;=\; \int_{\mathit{ff}} \delta(v).$

In order to prove these equalities, it is sufficient to prove them for terms p and q of the form $\sum_{i=1}^{k} \int_{\phi_i} P_i(v)$ (with $k \geq 1$), because in the previous section we got rid of time shifts and conditions in closed terms. As an example, we prove Equations 3.2 and 3.3.

Proof of Equation 3.2. Ensure by means of α-conversion that $v \notin var(\phi \vee \psi)$.

$$(\phi :\to \sum_{i=1}^{k} \int_{\phi_i} P_i(v)) + (\psi :\to \sum_{i=1}^{k} \int_{\phi_i} P_i(v))$$
$$\stackrel{\text{TC2,C3}}{=} \sum_{i=1}^{k} \int_{\phi\wedge\phi_i} P_i(v) + \sum_{i=1}^{k} \int_{\psi\wedge\phi_i} P_i(v)$$
$$\stackrel{\text{TA3}}{=} \sum_{i=1}^{k} \int_{(\phi\vee\psi)\wedge\phi_i} P_i(v) \stackrel{\text{TC2,C3}}{=} \phi \vee \psi :\to \sum_{i=1}^{k} \int_{\phi_i} P_i(v).\ \square$$

Proof of Equation 3.3.

$$\sum_{i=1}^{k} \int_{\phi_i} P_i(v) + \int_{ff} \delta(v) \stackrel{\text{TA6}}{=} \sum_{i=1}^{k} \int_{\phi_i} P_i(v) + \int_{ff} P_1(v) \stackrel{\text{TA3}}{=} \sum_{i=1}^{k} \int_{\phi_i} P_i(v).\ \square$$

3.2 Reducing conditions to intervals

A finite collection of conditions is called a *partition* if for each valuation σ there is exactly one condition ϕ in this collection such that $\sigma \in [\phi]$.

Equation 3.6 (lifting equation) *If* $\{\phi_1, ..., \phi_n\}$ *is a partition, then*

$$\int_{\phi} \alpha(v) \cdot \sum_{i=1}^{n} (\phi_i :\to p_i) \ = \ \sum_{i=1}^{n} \int_{\phi\wedge\phi_i} \alpha(v) \cdot p_i.$$

Proof. In this deduction we implicitly apply Equation 3.1.

$$\int_{\phi} \alpha(v) \cdot \sum_{i=1}^{n} (\phi_i :\to p_i)$$
$$\stackrel{\text{TA3}}{=} \sum_{j=1}^{n} \int_{\phi\wedge\phi_j} \alpha(v) \cdot \sum_{i=1}^{n} (\phi_i :\to p_i) \qquad \text{because } \vee_j \phi_j = tt$$
$$\stackrel{\text{TC1,C3}}{=} \sum_{j=1}^{n} \int_{\phi\wedge\phi_j} \alpha(v) \cdot \sum_{i=1}^{n} (\phi_i \wedge \phi_j :\to p_i)$$
$$\stackrel{\text{Eq 3.3}}{=} \sum_{j=1}^{n} \int_{\phi\wedge\phi_j} \alpha(v) \cdot (\phi_j :\to p_j) \qquad \text{because } \phi_i \wedge \phi_j = ff \text{ if } i \neq j$$
$$\stackrel{\text{TC1}}{=} \sum_{j=1}^{n} \int_{\phi\wedge\phi_j} \alpha(v) \cdot p_j.\ \square$$

The constraint in the lifting equation that $\{\phi_1, ..., \phi_n\}$ is a partition is an essential ingredient, because without it we would get equalities like

$$\int_{\phi} \alpha(v) \cdot \int_{ff} \delta(v) \ = \ \int_{ff} \delta(v),$$
$$\int_{\phi} \alpha(v) \cdot (p + q) \ = \ \int_{\phi} \alpha(v) \cdot p + \int_{\phi} \alpha(v) \cdot q.$$

The following equation enables to reduce the collection of conditions $\{\phi_i\}$ in a term $\sum_i (\phi_i :\to p_i)$ to a partition.

Equation 3.7 $(\phi :\to p) + (\psi :\to q) \ =$

$$(\phi \wedge \psi :\to p + q) + (\phi \wedge \neg\psi :\to p) + (\neg\phi \wedge \psi :\to q) + (\neg\phi \wedge \neg\psi :\to \int_{ff} \delta(v)).$$

Proof.

$$(\phi :\to p) + (\psi :\to q)$$
$$\stackrel{\text{Eq 3.2}}{=} (\phi \wedge \psi :\to p) + (\phi \wedge \neg\psi :\to p) + (\phi \wedge \psi :\to q) + (\neg\phi \wedge \psi :\to q)$$
$$\stackrel{\text{C3}}{=} (\phi \wedge \psi :\to p + q) + (\phi \wedge \neg\psi :\to p) + (\neg\phi \wedge \psi :\to q).$$

According to Equation 3.3, we can add $\int_{ff} \delta(v) \stackrel{\text{TC2}}{=} \neg\phi \wedge \neg\psi :\rightarrow \int_{ff} \delta(v)$. $\square$

We will use the lifting equation and the refinement lemma (see Section 2.1) to reduce conditions that parametrize integrals to the form *tt* or *ff* or $v = b$ or $v < b$ or $b < v$ or $b < v < b'$, with $v \notin var(b + b')$. In the sequel, such conditions will often be denoted by $v \in V$, where V represents an interval. For example, *tt* is denoted by $v \in \langle -\infty, \infty \rangle$, and *ff* by $v \in \emptyset$, and $b \leq v < b'$ by $v \in [b, b'\rangle$, etc.

3.3 Adapting deadlocks

Using the refinement lemma, conditions in deadlocks can be reduced to either *tt* or *ff* or $v = b$ by means of TA7 and the following two equations. Let $v \notin var(b + b')$.

Equation 3.8 $\int_{b<v} \delta(v) \;=\; \int_{tt} \delta(v)$.

Proof.
$$\int_{b<v} \delta(v) \stackrel{\text{TA3}}{=} \int_{b<v} \delta(v) + \int_{v=b+1} \delta(v) \stackrel{\text{TA7}}{=} \int_{b<v} \delta(v) + \int_{v<b+1} \delta(v) \stackrel{\text{TA3}}{=} \int_{tt} \delta(v).$$ $\square$

Equation 3.9 $\int_{b<v<b'} \delta(v) \;=\; b < b' :\rightarrow \delta(b')$.

This last equality can be deduced from TA3,7 and TC2.

Redundant deadlocks can be removed by means of TA4 and Equation 3.3 and the following equation, which can be deduced from TA3,4 and TC2.

Equation 3.10 $b \leq \sup(V) :\rightarrow (\int_{v\in V} P(v) + \delta(b)) \;=\; b \leq \sup(V) :\rightarrow \int_{v\in V} P(v)$.

3.4 Removing redundant variables

In a process term $\int_{v=b} a(v) \cdot p$ (with $v \notin var(b)$), the time variable v is 'redundant' in p, in the sense that it can only attain the value b. Occurrences of such redundant variables in p can be removed by the following equation.

Equation 3.11 $\int_{v=b} \alpha(v) \cdot p \;=\; \int_{v=b} \alpha(v) \cdot p[b/v]$.

This equation can be deduced from TC1,2 and C3 by induction on the *depth* of p (i.e. the length of the longest execution trace of p).

3.5 Removing double terms

In the reduction to normal form, we will delete double terms. Let $V \sim W$ denote the condition that $V \cup W$ is an interval. Axiom TA3 induces the following equation.

Equation 3.12 $V \sim W :\rightarrow (\int_{v\in V} P(v) + \int_{v\in W} P(v)) \;=\; V \sim W :\rightarrow \int_{v\in V\cup W} P(v)$.

Equation 3.12 is not always sufficient to reduce double terms. Namely, the equality

$$\int_{v<b} a(v) \cdot p + \int_{\phi} a(v) \cdot q \;=\; \int_{v\leq b} a(v) \cdot p + \int_{\phi} a(v) \cdot q$$

with $v \notin var(b)$, is sound if $\phi[b/v] = tt$ and $p[b/v] \underline{\leftrightarrow} q[b/v]$. However, this equation cannot be deduced from Equation 3.12, because p and q need not be of the same form. We introduce an extra equation to deal with this example.

In the reduction to normal form, this equation is only needed in case $p[b/v]$ and $q[b/v]$ are of the same form, which can be expressed by a condition as follows. Consider a term $\sigma(p)$. Reduce its bounds to normal form, using the axioms from BA, and replace subterms of the form $\int_{v=b} a(v) \cdot p'$ by $\int_{v=b} a(v) \cdot p'[b/v]$. The resulting process term will be denoted by $\sigma(p)*$. For p, q terms, let $\psi(p,q)$ denote a condition such that $\sigma \in [\psi(p,q)]$ if and only if $\sigma(p)* =_{\mathrm{AC}} \sigma(q)*$. Note that such a condition exists.

For p, q terms and b a bound with $v \notin var(b)$, the first example can be reduced by the following equation.

Equation 3.13 $\quad \phi[b/v] \wedge \psi(p[b/v], q[b/v]) :\rightarrow (\int_{v<b} a(v) \cdot p + \int_{\phi} a(v) \cdot q)$

$$= \quad \phi[b/v] \wedge \psi(p[b/v], q[b/v]) :\rightarrow (\textstyle\int_{v\leq b} a(v) \cdot p + \int_{\phi} a(v) \cdot q).$$

We also have a symmetric version of this equation, in order to reduce the process term $\int_{b<v} a(v) \cdot p + \int_{\phi} a(v) \cdot q$. We have two more symmetric equations to deal with the term $\int_{b_0<v<b_1} a(v) \cdot p + \int_{\phi} a(v) \cdot q$. These equations can be deduced from TA3 and TC2, C3 and Equation 3.11.

There is one more example which cannot be reduced neither by Equation 3.12 nor by Equation 3.13.

$$\textstyle\int_{v=b} a(v) \cdot p \;+\; \int_{\phi} a(v) \cdot q \quad \underline{\leftrightarrow} \quad \int_{\phi} a(v) \cdot q$$

if $\phi[b/v] = tt$ and $p \underline{\leftrightarrow} q[b/v]$. This second example can be reduced by the following equation.

Equation 3.14 $\quad \phi[b/v] \wedge \psi(p, q[b/v]) :\rightarrow (\int_{v=b} a(v) \cdot p + \int_{\phi} a(v) \cdot q)$

$$= \quad \phi[b/v] \wedge \psi(p, q[b/v]) :\rightarrow \textstyle\int_{\phi} a(v) \cdot q.$$

3.6 Construction of normal forms

We define an algorithm which reduces a basic term to a term which is called its *normal form*. This normal form is of the form $\sum_i (\phi_i :\rightarrow p_i)$, where $\{\phi_i\}$ is a partition and the p_i are basic terms. The algorithm uses the equations that have been deduced in the previous sections. Some of these equations are of the form $\phi :\rightarrow p = \phi :\rightarrow q$. In the algorithm, such equations are applied in the form $p = (\phi :\rightarrow q) + (\neg\phi :\rightarrow p)$.

We apply induction on depth. So suppose that we have already constructed normal forms for basic terms of depth $\leq n$, and let p be a basic term of depth $n+1$, of the form

$$\sum_{i\in I} \int_{\phi_i} \alpha_i(v) \cdot p_i \;+\; \sum_{j\in J} \int_{\phi_j} \alpha_j(v).$$

(In the basic induction step the sum over I is empty.) The p_i have depth $\leq n$, so by induction we already have constructed their normal forms $\sum_{k\in K_i}(\phi'_k :\rightarrow q_k)$. Replace the p_i by their normal forms and apply the lifting equation to obtain

$$\sum_{(i,k)\in I\times K_i} \int_{\phi_i\wedge\phi'_k} \alpha_i(v)\cdot q_k \;+\; \sum_{j\in J}\int_{\phi_j} \alpha_j(v).$$

According to the refinement lemma, $\phi_i \wedge \phi'_k$ is equivalent to a condition $\vee_{l\in L_k}\psi_l \wedge v \in V_l$, and ϕ_j is equivalent to a condition $\vee_{l\in L_j}\psi_l \wedge v \in V_l$, where $var(\psi_l)\cup var(V_l) \subseteq var(p)\backslash\{v\}$, and the V_l are either open or of the form $[b,b]$. Using TC2, we can reduce p to the form

$$\sum_{(i,k,l)\in I\times K_i\times L_k} (\psi_l :\rightarrow \int_{v\in V_l} \alpha_i(v)\cdot q_k) \;+\; \sum_{j\in J\ l\in L_j} (\psi_l :\rightarrow \int_{v\in V_l} \alpha_j(v)).$$

Reduce the conditions ψ_l to a partition by means of Equation 3.7. Reduce the bounds in the V_l to normal form, using the axioms in BA.

Next, we remove redundant variables. If a V_l is of the form $[b,b]$ and $v \in \mathit{fvar}(q_k)$, then apply Equation 3.11 to $\int_{v\in V_l} a_i(v)\cdot q_k$. Reduce the bounds in $q_k[b/v]$ to normal form again.

Next, we reduce deadlocks. We use the fact that the V_l are either open or of the form $[b,b]$.

- Reduce deadlocks $\int_{v\in V}\delta(v)\cdot q$ to the form $\int_{v\in V}\delta(v)$.
- Apply TA7 and Equations 3.8 and 3.9 in order to reduce conditions in deadlocks to either *tt* or *ff* or $v\in[b,b]$.
- Reduce terms $\int_{\mathit{ff}} P(v)$ to $\int_{\mathit{ff}}\delta(v)$.
- Remove redundant deadlocks using TA4 and Equations 3.3 and 3.10.

Finally, we remove double terms. First apply Equation 3.13, and then Equations 3.12 and 3.14 to each pair $\int_{v\in V} a(v)\cdot q \;+\; \int_{v\in W} a(v)\cdot q'$ and $\int_{v\in V} a(v) \;+\; \int_{v\in W} a(v)$. The result is the normal form of p.

3.7 The main theorem

Since each term is equal to a basic term, it follows that each term is equal to a normal form. Let $p\in\mathcal{T}^{cl}$ have normal form $\Sigma_i(\phi_i :\rightarrow p_i)$. The construction of normal forms ensures that $var(\phi_i)\cup \mathit{fvar}(p_i) \subseteq \mathit{fvar}(p) = \emptyset$. In particular, each ϕ_i is equal to either *tt* or *ff*, so we can reduce the normal form of p to a time-closed term by applying Equations 3.4 and 3.5.

We prove that bisimilar time-closed normal froms are equal modulo AC. First, we formulate two lemmas which are needed in the proof of the main theorem.

Lemma 3.15 *Let p and q be subterms of normal forms. If $p[r/v]* =_{\mathrm{AC}} q[r/v]*$ for infinitely many $r\in Time_0$, then $p =_{\mathrm{AC}} q$.*

The proof of this lemma, which is technical and straightforward, is presented in the appendix.

Lemma 3.16 *Let $\int_{v\in V} a(v)\cdot p$ be a normal form. Then $p[r/v]*$ is a normal form for all $r \in V \cap \mathit{Time}_0$.*

This lemma can be proved by showing that the construction to normal form reduces $p[t/v]$ to $p[t/v]*$. It is left to the reader to check that this is indeed the case.

Theorem 3.17 *If time-closed normal forms p and q are bisimilar, then $p =_{\mathrm{AC}} q$.*

Proof. We apply induction on the depth of p and q. First, let

$$p =_{\mathrm{AC}} \sum_{i\in I}\int_{v\in V_i}\alpha_i(v), \qquad q =_{\mathrm{AC}} \sum_{j\in J}\int_{v\in W_j}\alpha'_j(v).$$

Fix an $i \in I$. First, let $\alpha_i \in A$. Then for $t \in V_i$ we have $p \xrightarrow{\alpha_i(t)} \surd$. Since $p \underline{\leftrightarrow} q$, for each $t \in V_i$ there is a $j(t) \in J$ with $t \in W_{j(t)}$ and $\alpha_i = \alpha'_{j(t)}$. In the reduction to normal form Equation 3.12 has been applied, so the intervals $W_{j(t)}$ for $t \in V_i$ have been united to one interval. Hence, there is a unique $j \in J$ with $V_i \subseteq W_j$ and $\alpha_i = \alpha'_j$. Similarly, for this j there is a unique $i(j) \in I$ with $W_j \subseteq V_{i(j)}$ and $\alpha'_j = \alpha_{i(j)}$. Since $V_i \subseteq W_j \subseteq V_{i(j)}$ and $\alpha_i = \alpha'_j = \alpha_{i(j)}$, Equation 3.12 yields $i(j) = i$, so V_i and W_j coincide.

Next, let $\alpha_i = \delta$. The adaptation of deadlocks in the reduction of normal forms ensures that $v \in V_i$ is of the form $v < r$ or tt. Since Equation 3.10 has been applied in the reduction to normal form, the ultimate delay of p is determined by the summand $\int_{v<r}\delta(v)$ or $\int_{tt}\delta(v)$. Since p and q are bisimilar, it follows that q must contain this same summand.

Suppose that the theorem has been proved for depth $\leq n$. Let

$$p =_{\mathrm{AC}} \sum_{i\in I}\int_{v\in V_i} a_i(v)\cdot p_i + p', \qquad q =_{\mathrm{AC}} \sum_{j\in J}\int_{v\in W_j} a'_j(v)\cdot q_j + q',$$

where the p_i and q_j have depth n and p' and q' have depth $\leq n$. Since $p \underline{\leftrightarrow} q$, it follows that $p' \underline{\leftrightarrow} q'$ and thus by the induction hypothesis $p' =_{\mathrm{AC}} q'$.

Fix an $i \in I$. The terms p and q are bisimilar and are basic terms, so for each $t \in V_i$ there is a $j(t) \in J$ with $t \in W_{j(t)}$ and $a_i = a'_{j(t)}$ and $t \gg p_i[t/v] \underline{\leftrightarrow} t \gg q_{j(t)}[t/v]$. Since normal forms are basic terms, which means that they have ascending time stamps, it follows that $p_i[t/v] \underline{\leftrightarrow} t \gg p_i[t/v] \underline{\leftrightarrow} t \gg q_{j(t)}[t/v] \underline{\leftrightarrow} q_{j(t)}[t/v]$

First, assume that V_i contains more than one point. Let $J' \subseteq J$ be the collection of j for which $a'_j = a_i$ and $q_j =_{\mathrm{AC}} p_i$, and define $W_{J'} := \cup_{j\in J'} W_j$. We show that $V_i \backslash W_{J'}$ is empty.

1. $V_i \backslash W_{J'}$ contains only a finite number of time numbers in Time_0.

   Suppose not. For each $t \in V_i\backslash W_{J'}$ we have $j(t) \notin J'$, so then there is an infinite subset R of $(V_i\backslash W_{J'}) \cap \mathit{Time}_0$ and a $j_0 \in J\backslash J'$ such that $j(r) = j_0$ for all $r \in R$. Since $p_i[r/v] \underline{\leftrightarrow} q_{j_0}[r/v]$, the induction hypothesis together with Lemma 3.16 yield $p_i[r/v]* =_{\mathrm{AC}} q_{j_0}[r/v]*$ for $r \in R$. Since R is infinite, Lemma 3.15 implies $p_i =_{\mathrm{AC}} q_{j_0}$. Then $j_0 \in J'$, which is a contradiction.

2. $V_i \backslash W_{J'}$ does not contain time numbers in $\mathit{Time} \backslash \mathit{Time}_0$.

   Suppose that $V_i \backslash W_{J'}$ does contain a time number $t \notin \mathit{Time}_0$. Since $V_i \backslash W_{J'}$ consists of intervals that have bounds in Time_0, apparently it contains the time numbers in an interval $\langle r, s \rangle$ with $r, s \in \mathit{Time}_0$ and $r < t < s$. Then the time numbers $(r + (n-1)s)/n$ are distinct in $\langle r, s \rangle \cap \mathit{Time}_0$ for $n = 1, 2, \ldots$, so $V_i \backslash W_{J'}$ contains infinitely many time numbers in Time_0; contradiction.

3. $V_i \backslash W_{J'}$ is empty.

   Suppose not, so there exists an $r \in (V_i \backslash W_{J'}) \cap \mathit{Time}_0$. Since this set is finite, and since V_i contains more than one point, there is a $j \in J'$ such that W_j has lower bound or upper bound r. Since $q_j =_{\mathrm{AC}} p_i$ and $p_i[r/v] \underline{\leftrightarrow} q_{j(r)}[r/v]$, it follows that $q_j[r/v] \underline{\leftrightarrow} q_{j(r)}[r/v]$. Then the induction hypothesis together with Lemma 3.16 yield $q_j[r/v]* =_{\mathrm{AC}} q_{j(r)}[r/v]*$. Since Equation 3.13 has been applied to the pair

   $$\int_{v \in W_{j(r)}} a_i(v) \cdot q_{j(r)} \;+\; \int_{v \in W_j} a_i(v) \cdot q_j,$$

   we have $r \in W_j$, which is a contradiction.

Hence, $V_i \backslash W_{J'} = \emptyset$, or in other words, $V_i \subseteq W_{J'}$. Then Equation 3.12 implies that there is a unique $j \in J'$ with $V_i \subseteq W_j$.

Similarly for this j there is an $i(j) \in I$ with $a'_j = a_{i(j)}$ and $q_j =_{\mathrm{AC}} p_{i(j)}$ and $W_j \subseteq V_{i(j)}$. Since Equation 3.12 has been applied, it follows that $i(j) = i$. Hence, V_i coincides with W_j.

Next, assume that V_i equals $[t, t]$. If $W_{j(t)}$ contains more than one point, then we have just proved that there is an $i(t) \in I$ with $a_{i(t)} = a'_{j(t)}$, and $V_{i(t)}$ and $W_{j(t)}$ coincide, and $p_{i(t)} =_{\mathrm{AC}} q_{j(t)}$. Since $V_{i(t)}$ and $W_{j(t)}$ coincide, it follows that $t \in V_{i(t)}$. And $p_{i(t)} =_{\mathrm{AC}} q_{j(t)}$ together with $p_i[t/v] \underline{\leftrightarrow} q_{j(t)}[t/v]$ implies that $p_{i(t)}[t/v] \underline{\leftrightarrow} p_i[t/v]$. Then the induction hypothesis together with Lemma 3.16 give $p_{i(t)}[t/v]* =_{\mathrm{AC}} p_i[t/v]*$. Since Equation 3.14 has been applied to the pair

$$\int_{v \in V_{i(t)}} a_i(v) \cdot p_{i(t)} \;+\; \int_{v \in V_i} a_i(v) \cdot p_i,$$

the term $\int_{v \in V_i} a_i(v) \cdot p_i$ should not be there at all. This is a contradiction, so it follows that $W_{j(t)}$ equals $[t, t]$.

Since Equation 3.11 has been applied, $p_i =_{\mathrm{AC}} p_i[t/v]*$ and $q_{j(t)} =_{\mathrm{AC}} q_{j(t)}[t/v]*$. Induction and Lemma 3.16 yield $p_i[t/v]* =_{\mathrm{AC}} q_{j(t)}[t/v]*$, so $p_i =_{\mathrm{AC}} q_{j(t)}$. □

Corollary 3.18 *The axiomatization of BPA$\delta\rho$I is complete for time-closed processes modulo bisimulation equivalence.*

Proof. Let $p, q \in \mathcal{T}^{cl}$ be bisimilar. Each step in the reduction to normal form can be deduced from the axioms, so $p = p{\downarrow}$ and $q = q{\downarrow}$. Then $p{\downarrow} \underline{\leftrightarrow}\, p \,\underline{\leftrightarrow}\, q \,\underline{\leftrightarrow}\, q{\downarrow}$, so Theorem 3.17 yields $p{\downarrow} =_{\mathrm{AC}} q{\downarrow}$. Hence $p = q$. □

Corollary 3.19 *Bisimulation equivalence between time-closed processes in BPA$\delta\rho$I is decidable.*

3.8 An example

The normal form of a process term can be much larger than the term itself. For example, consider the term

$$\int_{0<v<5} a(v)\cdot(\int_{7-4v<w<5-v} b(w) \;+\; \int_{3<w<\frac{17}{6}+\frac{1}{3}v} b(w)).$$

Its normal form can be deduced from Figure 1. The lines that are drawn there intersect for $v\in\{\frac{1}{2},\frac{2}{3},\frac{25}{26},1,\frac{7}{5},\frac{13}{8},2,\frac{5}{2},3,\frac{17}{4}\}$. Thus we get the following normal form:

$$\begin{array}{llll}
 & \int_{0<v\le\frac{1}{2}} a(v)\cdot\delta(v) & + & \int_{\frac{1}{2}<v\le\frac{2}{3}} a(v)\cdot\int_{3<w<\frac{17}{6}+\frac{1}{3}v} b(w)\\
+ & \multicolumn{3}{l}{\int_{\frac{2}{3}<v\le\frac{25}{26}} a(v)\cdot(\int_{3<w<\frac{17}{6}+\frac{1}{3}v} b(w)+\int_{7-4v<w<5-v} b(w))}\\
+ & \int_{\frac{25}{26}<v\le 1} a(v)\cdot\int_{3<w<5-v} b(w) & + & \int_{1\le v\le\frac{7}{5}} a(v)\cdot\int_{7-4v<w<5-v} b(w)\\
+ & \int_{\frac{7}{5}\le v\le\frac{13}{8}} a(v)\cdot\int_{v<w<5-v} b(w) & + & \int_{\frac{13}{8}\le v<2} a(v)\cdot\int_{v<w<\frac{17}{6}+\frac{1}{3}v} b(w)\\
+ & \multicolumn{3}{l}{\int_{2\le v<\frac{5}{2}} a(v)\cdot(\int_{v<w<5-v} b(w)+\int_{3<w<\frac{17}{6}+\frac{1}{3}v} b(w))}\\
+ & \int_{\frac{5}{2}\le v\le 3} a(v)\cdot\int_{3<w<\frac{17}{6}+\frac{1}{3}v} b(w) & + & \int_{3\le v<\frac{17}{4}} a(v)\cdot\int_{v<w<\frac{17}{6}+\frac{1}{3}v} b(w)\\
+ & \int_{\frac{17}{4}\le v<5} a(v)\cdot\delta(v). & &
\end{array}$$

4 Parallelism and Synchronization

We introduce parallelism, synchronization and encapsulation, resulting in the theory ACPρI. We extend the syntax with the parallel merge $\|$, and we add the auxiliary operators left merge $⌊\!⌊$ and communication merge $|$, which allow the definition of $\|$ in finitely many axioms. Moreover, we add the encapsulation operator ∂_H.

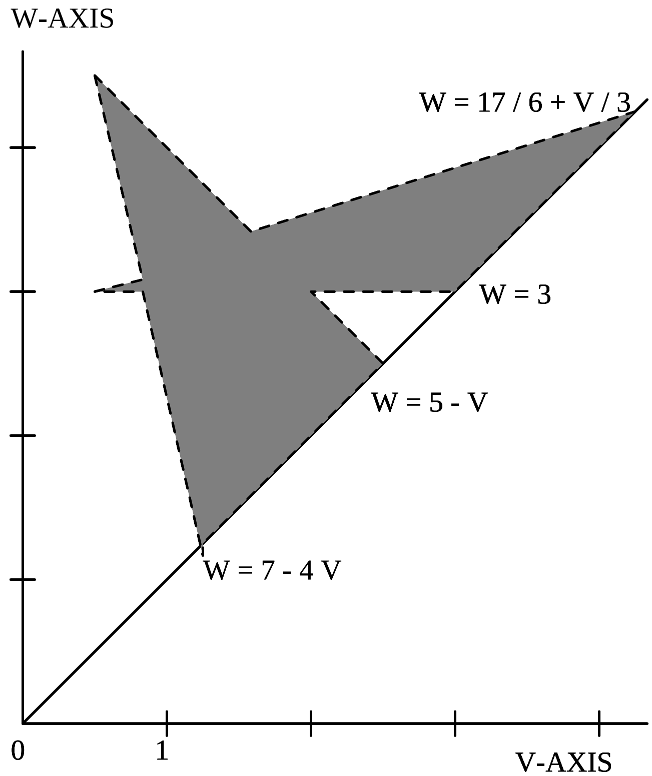
Assume a commutative communication function $|: A\cup\{\delta\}\times A\cup\{\delta\}\longrightarrow A\cup\{\delta\}$ which is commutative and associative and has δ as zero element. Communication between timed actions can only take place if they happen simultaneously. So if $a|a'=c$, then

$$\begin{array}{rcl} a(1)|a'(1) & = & c(1),\\ a(1)|a'(2) & = & \delta(1). \end{array}$$

In untimed ACP, $a ⌊\!⌊ p = a\cdot p$. However, in the real-time setting this definition would cause the process $a(1)\|a'(2)$ to have a deadlock:

$$a(1)\|a'(2) = a(1)⌊\!⌊ a'(2) + a'(2)⌊\!⌊ a(1) = a(1)\cdot a'(2) + a'(2)\cdot\int_{f\!f}\delta(v).$$

Figure 1: Graphical description of a process term



In order to avoid such deadlocks, the definition of the left merge is adapted in such a way that a process $p ⌊\!⌊ q$ can only idle as long as both p and q can idle. So for example:

$$\begin{aligned} a(1) ⌊\!⌊ a'(2) &= a(1) \cdot a'(2), \\ a'(2) ⌊\!⌊ a(1) &= \delta(1). \end{aligned}$$

Note that this definition induces $a(1) \| a'(2) = a(1) \cdot a'(2)$.

For each subset H of A, the encapsulation operator ∂_H is defined on $A \cup \{\delta\}$ by $\partial_H(\alpha) = \delta$ if $\alpha \in H \cup \{\delta\}$ and $\partial_H(a) = a$ if $a \in A \backslash H$.

The constructs $\|$ and $⌊\!⌊$ and $|$ and ∂_H are added to the notion of a process term, that is, the BNF grammar is extended with

$$p \| p \mid p ⌊\!⌊ p \mid p | p \mid \partial_H(p).$$

The notions free variables and substitutions extend to these operators as expected.

4.1 Operational semantics for ACPρI

Table 5 contains the action rules for the operators $\|, |$, $⌊\!⌊$ and ∂_H. Once more, bounds may contain time numbers t from *Time*. The rules are within the format of Bloom and Vaandrager, so bisimulation equivalence is a congruence for the added operators as well.

Furthermore, ACPρI is a conservative extension of BPA$\rho\delta I$, which means that the operational semantics of BPA$\rho\delta I$ terms is not influenced by the extra action rules for communication and encapsulation. This follows from the fact that the action rules of BPA$\rho\delta I$ are all *source-dependent*, and that the extra action rules of ACPρI all have a new function symbol in the left-hand side of their conclusion. See Fokkink and Verhoef (1995) for the definitions and a proof of this result.

Table 5: Additional action rules for ACPρI

$$\frac{x \xrightarrow{a(t)} x' \quad U_t(y)}{x\|y \xrightarrow{a(t)} x'\|(t \gg y) \quad y\|x \xrightarrow{a(t)} (t \gg y)\|x' \quad x\mathbin{\|\!\_} y \xrightarrow{a(t)} x'\|(t \gg y)}$$

$$\frac{x \xrightarrow{a(t)} \surd \quad U_t(y)}{x\|y \xrightarrow{a(t)} t \gg y \quad y\|x \xrightarrow{a(t)} t \gg y \quad x\mathbin{\|\!\_} y \xrightarrow{a(t)} t \gg y}$$

If $a|a' = c \neq \delta$, then

$$\frac{x \xrightarrow{a(t)} x' \quad y \xrightarrow{a'(t)} y'}{x\|y \xrightarrow{c(t)} x'\|y' \quad x|y \xrightarrow{c(t)} x'\|y'} \qquad \frac{x \xrightarrow{a(t)} \surd \quad y \xrightarrow{a'(t)} \surd}{x\|y \xrightarrow{c(t)} \surd \quad x|y \xrightarrow{c(t)} \surd}$$

$$\frac{x \xrightarrow{a(t)} \surd \quad y \xrightarrow{a'(t)} y'}{x\|y \xrightarrow{c(t)} y' \quad y\|x \xrightarrow{c(t)} y' \quad x|y \xrightarrow{c(t)} y' \quad y|x \xrightarrow{c(t)} y'}$$

$$\frac{x \xrightarrow{a(t)} \surd \quad a \notin H}{\partial_H(x) \xrightarrow{a(t)} \surd} \qquad \frac{x \xrightarrow{a(t)} x' \quad a \notin H}{\partial_H(x) \xrightarrow{a(t)} \partial_H(x')}$$

$$\frac{U_t(x) \quad U_t(y)}{U_t(x\|y) \quad U_t(x\mathbin{\|\!\_} y) \quad U_t(x|y)} \qquad \frac{U_t(x)}{U_t(\partial_H(x))}$$

4.2 Axioms for ACPρI

The axiomatization for ACPρI consists of the (old) axioms in Table 4, together with the (new) axioms in Table 6. Some of these new axioms contain the construct $U_b(p)$, which represents a condition which results to true under a valuation σ if and only if $U_{\sigma(b)}(\sigma(p))$. At the end of Table 6, six axioms U1-6 are added which enable to reduce the construct $U_b(p)$ to a condition of the form as defined in Section 2.1. Table 6 uses the abbreviation $U_b(\phi)$ for $U_b(\int_\phi \delta(v))$.

Table 6: Additional axioms for ACPρI

CM1		$x \parallel y = x \mathbin{⌊\!⌊} y + y \mathbin{⌊\!⌊} x + x \mid y$
TCM2	$v \notin fvar(y)$	$\int_\phi \alpha(v) \mathbin{⌊\!⌊} y = \int_{\phi \wedge U_v(y)} \alpha(v) \cdot y + \int_{U_v(\phi) \wedge U_v(y)} \delta(v)$
TCM3	$v \notin fvar(y)$	$(\int_\phi \alpha(v) \cdot x) \mathbin{⌊\!⌊} y = \int_{\phi \wedge U_v(y)} \alpha(v) \cdot (x \parallel y) + \int_{U_v(\phi) \wedge U_v(y)} \delta(v)$
CM4		$(x + y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} z + y \mathbin{⌊\!⌊} z$
TCF		$\int_\phi \alpha(v) \mid \int_\psi \alpha'(v) = \int_{\phi \wedge \psi} (\alpha \mid \alpha')(v) + \int_{U_v(\phi) \wedge U_v(\psi)} \delta(v)$
TCM5		$(\int_\phi \alpha(v) \cdot x) \mid \int_\psi \alpha'(v) = \int_{\phi \wedge \psi} (\alpha \mid \alpha')(v) \cdot x + \int_{U_v(\phi) \wedge U_v(\psi)} \delta(v)$
TCM6		$\int_\phi \alpha(v) \mid \int_\psi \alpha'(v) \cdot y = \int_{\phi \wedge \psi} (\alpha \mid \alpha')(v) \cdot y + \int_{U_v(\phi) \wedge U_v(\psi)} \delta(v)$
TCM7		$(\int_\phi \alpha(v) \cdot x) \mid \int_\psi \alpha'(v) \cdot y = \int_{\phi \wedge \psi} (\alpha \mid \alpha')(v) \cdot (x \parallel y) + \int_{U_v(\phi) \wedge U_v(\psi)} \delta(v)$
CM8		$(x + y) \mid z = x \mid z + y \mid z$
CM9		$x \mid (y + z) = x \mid y + x \mid z$
TD1		$\partial_H(\int_\phi \alpha(v)) = \int_\phi \partial_H(\alpha)(v)$
TD2		$\partial_H(\int_\phi \alpha(v) \cdot x) = \int_\phi \partial_H(\alpha)(v) \cdot \partial_H(x)$
D3		$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$
U1		$U_b(\int_\phi P(v)) = U_b(\int_\phi \delta(v))$
U2	$v \notin fvar(b')$	$U_b(\int_{v<b'} \delta(v)) = b < b'$
U3		$U_b(\int_{tt} \delta(v)) = tt$
U4		$U_b(\int_{ff} \delta(v)) = ff$
U5		$U_b(x + y) = U_b(x) \vee U_b(y)$
U6		$U_b(\phi :\rightarrow x) = \phi \wedge U_b(x)$

In the sequel, $p = q$ means that equality between these terms can be derived from the equational theory of ACPρI. We extend our decidability result to ACPρI by showing how to eliminate the new operators from the syntax, using these axioms.

Similarly as in Proposition 2.5 we can deduce that if two time-closed terms are provably equal in the conditional axiom system, then they are bisimilar.

The following Proposition 4.1 says that the axioms of ACPρI are sufficient to eliminate the communication and encapsulation operators from the syntax. Godskesen and Larsen (1992) provided a rigorous proof that time dependencies are essential in order to obtain such a theorem in a timed setting. (Aceto and Murphy (1993) proposed the notion of 'ill-timed' traces, in order to obtain an expansion theorem for the merge in the absence of time dependencies.)

Proposition 4.1 *For each ACPρI term p, there is a BPA$\delta\rho I$ term p' such that $p = p'$.*

Proof. Let p_0 and p_1 denote two BPA$\delta\rho I$ terms. First, suppose that p is of the form $p_0 \mathbin{\|\!\_} p_1$. We show how to eliminate $\mathbin{\|\!\_}$ from this term, by induction on the sizes of p_0 and p_1. Previously, we have shown for terms that do not contain communication nor encapsulation operators, that time shifts can be deleted. Hence, only the following cases need to be considered.

$$\begin{array}{rcll} \int_\phi \alpha(v) \mathbin{\|\!\_} p_1 & = & \int_{\phi \wedge U_v(p_1)} \alpha(v) \cdot p_1 + \int_{U_v(\phi) \wedge U_v(p_1)} \delta(v) & v \notin \mathit{fvar}(p_1) \\ (\int_\phi \alpha(v) \cdot q) \mathbin{\|\!\_} p_1 & = & \int_{\phi \wedge U_v(p_1)} \alpha(v) \cdot (q \| p_1) + \int_{U_v(\phi) \wedge U_v(p_1)} \delta(v) & v \notin \mathit{fvar}(p_1) \\ (q + q') \mathbin{\|\!\_} p_1 & = & q \mathbin{\|\!\_} p_1 + q' \mathbin{\|\!\_} p_1. & \end{array}$$

Similarly, we can delete $|$ from $p_0 | p_1$, using the TCM axioms from Table 6. Since $p_0 \| p_1 = p_0 \mathbin{\|\!\_} p_1 + p_1 \mathbin{\|\!\_} p_0 + p_0 | p_1$, the case $p_0 \| p_1$ reduces to the previous two cases. Finally, the ∂_H can be deleted from $\partial_H(p_0)$ by means of the TD axioms.

Next, we deal with the general case, where may p contain several communication and encapsulation operators. These operators can be deleted one by one, by considering subterms of p of the form $p_0 \| p_1$ or $p_0 \mathbin{\|\!\_} p_1$ or $p_0 | p_1$ or $\partial_H(p_0)$. □

In the previous section we have seen that by means of the axioms in Table 4 we can decide bisimulation equivalence between time-closed terms that do not contain communication nor encapsulation operators. Together with Proposition 4.1, it follows that the equational theory of ACPρI decides bisimulation equivalence between time-closed terms.

Corollary 4.2 *The equational theory of ACPρI is complete for time-closed processes modulo bisimulation equivalence.*

Corollary 4.3 *Bisimulation equivalence between time-closed processes in ACPρI is decidable.*

5 Related Work

Recently, many process algebras have been extended with some notion of time. Most timed process algebras are based on relative time, in which the time stamp of an action refers to the point in time when the previous action was executed. This paper focused on absolute time, which means that the time stamp of an action $a(t)$ refers to the start time of the entire process (time zero). A real time version of ACP with relative time was proposed in Baeten and Bergstra (1991), in which square brackets are used to denote relative time; so $a[1] \cdot b[2]$ corresponds with the (absolute time) expression $a(1) \cdot b(3)$. Without any complications our results can be translated to ACP with relative time

We consider timed process algebras with a notion of integration and time dependencies. For example, we do not consider the work of Holmer, Larsen, and Wang (1991) on decidability in real time CCS, because their algebra does not incorporate time dependencies.

5.1 Timed CCS

Wang (1991) introduced the construct $a@v.p$, which executes a at some time t, after which it evolves into $p[t/v]$. His calculus is based on relative time, and unlike timed ACP, consecutive actions can be executed at the same point in time. The construct $a@v.p$ is equivalent to the expression $\int_{0\leq v} a[v] \cdot p$ in BPA with relative time.

Chen (1992) presented a generalization of Wang's construct $a@v.p$, namely $a(v)|_b^{b'}.p$, which executes a at some point in time $t \in [b, b']$, after which it evolves into $p[t/v]$. This construct is similar to the expression $\int_{b\leq v\leq b'} a[v] \cdot p$ in BPA with relative time. The bounds b and b' allow to express time dependencies.

Chen obtained a decidability result by introducing for every pair of processes p, q a first order formula $WC(p,q)$ which is the least condition such that p and q are bisimilar. Decidability follows from the decidability of the first order theory of the underlying time domain, according to Tarski (1951).

Chen (1993) introduced an axiom system with conditions. Derivations are relative to some condition; if two process terms p, q, possibly containing free time variables, are equal under the condition ϕ, then $\phi \vdash p = q$. He shows that $WC(p,q)$ can be expressed by a condition, and that $WC(p,q) \vdash p = q$ is derivable, which induces that his axiom system is effective. In Chen's setting it is not possible to mix conditions through the terms. In order to explain the difference with our axiomatization, we rephrase axiom TC1 as a conditional proof rule:

$$\frac{\phi \vdash x = y}{tt \vdash \int_\phi \alpha(v) \cdot x = \int_\phi \alpha(v) \cdot y}$$

A derivation starting from a term $\int_\phi \alpha(v) \cdot x$ where ϕ is used 'deep down' in x, gives rise to a proof tree, while in our setting derivations are always equational.

5.2 Timed automata

Alur and Dill (1994) proposed an extension of Büchi and Muller automata with time. Transitions are supplied with time constraints on 'clock variables', and while executing a transition, a clock can be set back to zero. A trace is accepted by a timed automaton if its transitions are performed at times that all clocks satisfy their constraints. Furthermore, accepted traces have to satisfy required fairness constraints, and Zeno behaviour is excluded from timed automata, i.e. traces are only accepted if they progress beyond any moment in time. The fairness restrictions, the non-Zeno requirement and the fact that only infinite traces are considered, are obstacles for the translation between timed automata and real time ACP with recursion. However, if these restrictions are discarded, then the classes of timed automata corresponds with a subalgebra of real time ACP with recursion which allows an elimination theorem for the merge, see Fokkink (1993).

Čerāns (1992) introduced a more general notion of timed automata, which he calls timed graphs. For example, in his setting edges of automata are painted a colour, red or black, which determines whether or not idling is allowed when the edge becomes enabled. Čerāns proved that bisimulation equivalence is decidable for timed graphs. This result is incomparable with ours, because his timed algebra is so different.

Acknowledgements

A great number of people provided useful comments. Special thanks go to Jos Baeten, Frits Vaandrager and an anonymous referee, who suggested many substantial improvements.

6 Appendix: Three Proofs

Refinement Lemma. Fix a time variable v. Each condition ϕ is equal to a condition of the form $\vee_i(\phi_i \wedge \phi_i')$, where

- $var(\phi_i) \subseteq var(\phi)\backslash\{v\}$,
- ϕ_i' is of the form $v = b$ or $v < b$ or $b < v$ or $b < v < b'$, with $var(b+b') \subseteq var(\phi)\backslash\{v\}$.

Proof sketch. First, rewrite ϕ to a condition of the form $\vee_i\psi_i$, with each ψ_i of the form $\wedge_j(b_j < b_j') \;\wedge\; \wedge_k\,(c_k = c_k')$. Reduce the bounds in ψ_i to normal form, i.e. to the form $r_1 \cdot v_1 + ... + r_l \cdot v_l + s$. In each (in)equality, collect factors $r \cdot v$ at one side, and collect the remaining summands of the bounds on the other side, such that either v is deleted from the (in)equality, or it takes the form $r \cdot v = b$ or $r \cdot v < b$, with $r \neq 0$ and $v \notin var(b)$. In the latter case, replace the (in)equality by $v = (1/r) \cdot b$ or by $v < (1/r) \cdot b$ if $r > 0$ or by $(1/r) \cdot b < v$ if $r < 0$. Thus we can reduce each ψ_i to an equivalent condition ψ_i' of the form

$$\psi \;\wedge\; \bigwedge_{j\in J} b_j < v \;\wedge\; \bigwedge_{k\in K} v < c_k \;\wedge\; \bigwedge_{l\in L} v = d_l$$

where v does not occur in ψ, b_j, c_k, d_l. We show that such a ψ_i' is equivalent to a condition of the form $\vee_j(\phi_j' \wedge v \in V_j)$, with $v \notin var(\phi_j') \cup var(V_j)$.

First, suppose $L \neq \emptyset$. Fix an $l_0 \in L$ and put $d = d_{l_0}$. The following condition is equivalent to ψ_i'.

$$(\psi \wedge \bigwedge_{j\in J} b_j < d \wedge \bigwedge_{k\in K} d < c_k \wedge \bigwedge_{l\in L} d = d_l) \;\wedge\; v = d.$$

So we may assume $L = \emptyset$. If $J \neq \emptyset$ and $K \neq \emptyset$, then the following condition is equivalent to ψ_i'.

$$\bigvee_{(j,k)\in J\times K} (\psi \wedge \bigwedge_{j'\in J} b_{j'} \leq b_j \wedge \bigwedge_{k'\in K} c_k \leq c_{k'} \wedge b_j < v < c_k).$$

Similarly, we can find suitable conditions equivalent to ψ_i' if J or K is empty. □

Completeness of CA. If $[\phi] = [\phi']$, then $\phi = \phi'$.

Proof sketch. We apply induction on the number of variables that occur in ϕ and ϕ'. If this number is zero, then the proposition is trivial, since then both ϕ and ϕ' reduce to either *tt* or *ff*. So assume that we have proved the case for n variables, and let ϕ and

ϕ' contain $n+1$ variables. Fix a variable v that occurs in ϕ or in ϕ'. According to the refinement lemma, we have

$$\phi = \bigvee_i \psi_i \wedge (v \in V_{i1} \vee ... \vee v \in V_{im_i}) \qquad \phi' = \bigvee_i \psi_i \wedge (v \in W_{i1} \vee ... \vee v \in W_{in_i})$$

where v does not occur in the ψ_i, V_{ij}, W_{ij}, and $\{\psi_i\}$ is a partition. Furthermore, ensure that under condition ψ_i both the V_{ij} and the W_{ij} are pairwise disjoint and non-empty. Moreover, ensure that if $\sigma \in [\psi_i]$, then the elements in $\sigma(V_{ij})$ and in $\sigma(W_{ij})$ are smaller than the elements in $\sigma(V_{ij+1})$ and in $\sigma(W_{ij+1})$ respectively.

Consider a ψ_i, and suppose that $\sigma \in [\psi_i]$. Since $\sigma(\phi) = \sigma(\phi')$, and since $\{\psi_i\}$ is a partition, it follows that $\sigma(v \in V_{i1} \vee ... \vee v \in V_{im_i}) = \sigma(v \in W_{i1} \vee ... \vee v \in W_{in_i})$. The $\sigma(V_{ij})$ and the $\sigma(W_{ij})$ are pairwise disjoint and of increasing order, so $m_i = n_i$ and $\sigma(V_{ij}) = \sigma(W_{ij})$ for all j. So if V_{ij} has lower bound b_j and upper bound b'_j, and let W_{ij} have lower bound c_j and upper bound c'_j, then $\sigma(\psi_i \wedge b_j = c_j \wedge b'_j = c'_j) = \sigma(\psi_i)$. This equality holds as well if $\sigma(\psi_i)$ results to false, so the induction hypothesis yields $(\psi_i \wedge b_j = c_j \wedge b'_j = c'_j) = \psi_i$. Hence, $\psi_i \wedge v \in V_{ij} \;=\; \psi_i \wedge v \in W_{ij}$. This holds for all i and j, so

$$\bigvee_i \psi_i \wedge (v \in V_{i1} \vee ... \vee v \in V_{im_i}) \;=\; \bigvee_i \psi_i \wedge (v \in W_{i1} \vee ... \vee v \in W_{in_i}). \; \Box$$

Lemma 3.15. Let p and q be subterms of normal forms. If $p[r/v]* =_{\mathrm{AC}} q[r/v]*$ for infinitely many $r \in \mathit{Time}_0$, then $p =_{\mathrm{AC}} q$.

Proof. For a bound b, let $b\!\downarrow$ be its normal form. Note that if $b_0\!\downarrow \neq_{\mathrm{AC}} b_1\!\downarrow$, then there is at most one $r \in \mathit{Time}_0$ such that $b_0[r/v]\!\downarrow \; =_{\mathrm{AC}} b_1[r/v]\!\downarrow$. Let $\lhd b_0, b_1 \rhd\!\downarrow$ denote $\lhd b_0\!\downarrow, b_1\!\downarrow \rhd$.

We use induction on the depth of p and q. Let

$$p \;=_{\mathrm{AC}}\; \sum_i \int_{w \in V_i} a_i(w) \cdot p_i \;+\; \sum_j \int_{w \in W_j} \alpha_j(w),$$

$$q \;=_{\mathrm{AC}}\; \sum_k \int_{w \in V'_k} a'_k(w) \cdot q_k \;+\; \sum_l \int_{w \in W'_l} \alpha'_l(w).$$

Assume that $p \neq_{\mathrm{AC}} q$; we show that $p[r/v]* =_{\mathrm{AC}} q[r/v]*$ for only finitely many $r \in \mathit{Time}_0$. We distinguish two cases.

1. There is a j such that for all l we have $\int_{w \in W_j} \alpha_j(w) \neq_{\mathrm{AC}} \int_{w \in W'_l} \alpha'_l(w)$.

   Fix an l. If $\alpha_j \neq \alpha'_l$, then clearly $\int_{w \in W_j[r/v]\Downarrow} \alpha_j(w) \neq_{\mathrm{AC}} \int_{w \in W'_l[r/v]\Downarrow} \alpha'_l(w)$ for all r.

   So assume that $\alpha_j = \alpha'_l$. Then $W_j \neq_{\mathrm{AC}} W'_l$, so there is no more than one $r \in \mathit{Time}_0$ such that $W_j[r/v]\!\downarrow =_{\mathrm{AC}} W'_l[r/v]\!\downarrow$.

   It follows that the set $\{r \in \mathit{Time} \mid p[r/v]* =_{\mathrm{AC}} q[r/v]*\}$ is smaller or equal to the number of l's (and thus finite).

2. There is an i such that for all k we have $\int_{w\in V_i} a_i(w)\cdot p_i \neq_{\mathrm{AC}} \int_{w\in V_k'} a_k'(w)\cdot q_k$.

   Fix a k with $a_i = a_k'$. If $V_i \neq_{\mathrm{AC}} V_k'$, then it follows as in 1 that there is no more than one r such that

$$(\int_{w\in V_i} a_i(w)\cdot p_i)[r/v]* =_{\mathrm{AC}} (\int_{w\in V_k'} a_k'(w)\cdot q_k)[r/v]*\,.$$

   So assume that $p_i \neq_{\mathrm{AC}} q_k$. Then by the induction hypothesis there is only a finite number of r such that $p_i[r/v]* =_{\mathrm{AC}} q_k[r/v]*$. Furthermore, if V_i or V_k' is not of the form $[b,b]$, then there is no more than one r such that $V_i[r/v]\downarrow$ or $V_k'[r/v]\downarrow$ does have this form respectively.

   It follows that $\{r\in V\cap \mathit{Time}_0 \mid p[r/v]* =_{\mathrm{AC}} q[r/v]*\}$ is finite. □

References

[1] Aceto, L., and Murphy, D. (1993), On the ill-timed but well-caused, *in* "Proceedings, 4th Conference on Concurrency Theory (CONCUR'93), Hildesheim," (E. Best, Ed.), pp. 97–111, Lecture Notes in Computer Science, Vol. 715, Springer-Verlag, Berlin.

[2] Alur, R., and Dill, D. L. (1994), A theory of timed automata, *Theoret. Comput. Sci.*, **126**(2), 183–235.

[3] Baeten, J. C. M., and Bergstra, J. A. (1991), Real time process algebra, *Form. Asp. Comput.*, **3**(2), 142–188.

[4] Bloom, B., and Vaandrager, F. W. (1995), Unpublished manuscript on SOS rule formats for parametrized and state-bearing processes.

[5] Čerāns, K. (1992), Decidability of bisimulation equivalence for processes with parallel timers, Technical report, University of Latvia, Riga.

[6] Chang, C. C., and Keisler, H. J. (1990), "Model Theory," North-Holland, Amsterdam.

[7] Chen, L. (1992), An interleaving model for real-time systems, *in* "Proceedings, 2nd Symposium on Logical Foundations of Computer Science, Tver, Russia," (A. Nerode and M. Taitslin, Eds.), pp. 81–92, Lecture Notes in Computer Science, Vol. 620, Springer-Verlag, Berlin.

[8] Chen, L. (1993), Axiomatising real-timed processes, *in* "Proceedings, 9th Conference on Mathematical Foundations of Programming Semantics (MFPS'93), New Orleans," (S. Brooks, M. Main, A. Melton, M. Mislove and D. Schmidt, Eds.), pp. 215–229, Lecture Notes in Computer Science, Vol. 802, Springer-Verlag, Berlin.

[9] Fokkink, W. J. (1991), Normal forms in real time process algebra, Report CS-R9149, CWI, Amsterdam.

[10] Fokkink, W. J. (1993), An elimination theorem for regular behaviours with integration, *in* "Proceedings, 4th Conference on Concurrency Theory (CONCUR'93), Hildesheim," (E. Best, Ed.), pp. 432–446, Lecture Notes in Computer Science, Vol. 715, Springer-Verlag, Berlin.

[11] Fokkink, W. J., and Verhoef, C. (1995), Unpublished manuscript on conservative extension in operational semantics with types and variable binding.

[12] Godskesen, J. C., and Larsen, K. G. (1992), Real time calculi and expansion theorems, *in* "Proceedings, 12th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 12), New Delhi," (R. Shyamasundar, Ed.), pp. 302–315, Lecture Notes in Computer Science, Vol. 652, Springer-Verlag, Berlin.

[13] Holmer, U., Larsen, K. G., and Wang, Y. (1991), Deciding properties of regular real timed processes, *in* "Proceedings, 3rd Workshop on Computer Aided Verification (CAV'91), Aalborg," pp. 432–442, Lecture Notes in Computer Science, Vol. 575, Springer-Verlag, Berlin.

[14] Klusener, A. S. (1991a), Completeness in real time process algebra, *in* "Proceedings, 2nd Conference on Concurrency Theory (CONCUR'91), Amsterdam," (J.C.M. Baeten and J.F. Groote, Eds.), pp. 376–392, Lecture Notes in Computer Science, Vol. 527, Springer-Verlag, Berlin.

[15] Klusener, A. S. (1991b), Abstraction in real time process algebra, *in* "Proceedings, REX Workshop, Real Time: Theory in Practice, Mook," (J.W. de Bakker, C. Huizing, W.P. de Roever and G. Rozenberg, Eds.), pp. 325–352, Lecture Notes in Computer Science, Vol. 600, Springer-Verlag, Berlin.

[16] Klusener, A. S. (1992), The silent step in time, *in* "Proceedings, 3rd Conference on Concurrency Theory (CONCUR'92), Stony Brook," (W.R. Cleaveland, Ed.), pp. 421–435, Lecture Notes in Computer Science, Vol. 630, Springer-Verlag, Berlin.

[17] Klusener, A. S. (1993), "Models and Axioms for a Fragment of Real Time Process Algebra," PhD thesis, Eindhoven University of Technology, Eindhoven.

[18] Koppelberg, S. (1989), General theory of Boolean algebras, *in* "Handbook of Boolean Algebras, Volume 1," (J.D. Monk, Ed.), North-Holland, Amsterdam.

[19] Moller, F., and Tofts, C. (1990), A temporal calculus of communicating systems, *in* "Proceedings, 1st Conference on Concurrency Theory (CONCUR'90), Amsterdam," (J.C.M. Baeten and J.W. Klop, Eds.), pp. 401–415, Lecture Notes in Computer Science, Vol. 458, Springer-Verlag, Berlin.

[20] Schneider, S. A. (1995), An operational semantics for timed CSP, *Inform. Comput.*, **116**(2), 193–213.

[21] Stoughton, A. (1988), Substitution revisited, *Theoret. Comput. Sci.*, **59**, 317–325.

[22] TARSKI, A. (1951), "A New Decision Method for Elementary Algebra," University of California Press, Berkeley.

[23] WANG, Y. (1990), Real time behaviour of asynchronous agents, *in* "Proceedings, 1st Conference on Concurrency Theory (CONCUR'90), Amsterdam," (J.C.M. Baeten and J.W. Klop, Eds.), pp. 502–520, Lecture Notes in Computer Science, Vol. 458, Springer-Verlag, Berlin.

[24] WANG, Y. (1991), CCS + time = an interleaving model for real time systems, *in* "Proceedings, 18th International Colloquium on Automata, Languages and Programming (ICALP'91), Madrid," (J. Leach Albert, B. Monien, and M. Rodríguez, Eds.), pp. 217–228, Lecture Notes in Computer Science, Vol. 510, Springer-Verlag, Berlin.