1992

W.J. Fokkink, A.S. Klusener

Real time process algebra with prefixed integration

Computer Science/Department of Software Technology   Report CS-R9219 June

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

Real Time Process Algebra with Prefixed Integration

Willem Jan Fokkink
Steven Klusener
CWI
Department of Software Technology
P.O. Box 4079, 1009 AB Amsterdam,
The Netherlands

Abstract

Recently J.C.M. Baeten and J.A. Bergstra extended ACP with real time, resulting in a Real Time Process Algebra called ACPρ [BB91]. They introduced an equational theory together with an operational semantics which contains the notion of an idle transition, reflecting that a process can do nothing more than passing the time before performing a concrete action at a certain point in time. This idle transition corresponds nicely to our intuition, but it results in uncountably branching transition systems. Their paper does not contain a completeness result nor definitions for detailed proofs.

In this paper we give a more abstract operational semantics without the idle transition. Due to this simplification and by restricting to *prefixed* integration we can prove soundness and completeness. Furthermore, we will show that equality between process terms is decidable.

1985 Mathematics Subject Classification: 68Q10, 68Q40, 68Q55, 68Q60.
1982 CR Categories: D.1.3, D.3.1, D.4.1, F.3.1, F.3.2.
Key Words & Phrases: real time, process algebra, ACP, integration, SOS, normal form.
Note: The authors were partly sponsored by ESPRIT Basic Research Action 3006, CONCUR.

Introduction

Since real life communicating protocols often depend on time constraints, many existing process algebras have been extended with an explicit notion of time. See for instance [MT90],[Wan90] for an extension of CCS, [RR88] for an extension of CSP, [Gro90],[BB91] for an extension of ACP and [NS90] for an extension of a combination of CCS and ACP.

This paper is based on the approach of Baeten and Bergstra in [BB91]. Real Time Process Algebra concerns (closed) process terms, constructed from timed actions that consist of a symbolic action taken from an alphabet A and a time stamp taken from $[0, \infty]$. A timed action $a(t)$ denotes the process which executes an action a at time t, after which it terminates successfully. This results in identities that do not hold in standard Process Algebra ([BK84],[BW90]), such as

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

After doing $a(2)$ we have passed time 2, so in the remaining subterm $b(1) + c(3)$ the first alternative cannot be chosen anymore and therefore it may be removed.

Report CS-R9219
ISSN 0169-118X
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

In their paper Baeten and Bergstra have also introduced the advanced notion of integration, by which it can be expressed that an action occurs somewhere within a (dense) interval. For example, the process which executes an action a somewhere within the interval $\langle 0,1\rangle$ is denoted by $\int_{v\in\langle 0,1\rangle} a(v)$. In general their theory is undecidable, due to the fact that they allow integration over arbitrary subsets of $[0,\infty]$. In this paper we restrict ourselves to prefixed integration, which requires that every action having a time variable v as time stamp is directly preceded by the binding integral of v. Furthermore, integration is only allowed over intervals of which the bounds are linear expressions of variables.

We will focus on absolute time, i.e. every time stamp is interpreted from the start time of the entire process (time zero). It is also possible to consider process terms in relative time, where every time stamp refers to the point in time the previous action was executed. In [BB91] a translation from relative to absolute time has been given. Hence the results of this paper hold for relative time as well.

Operational semantics, bisimulation equivalence and equational theory

In this paper an operational semantics consists of a set of states and a transition relation. This relation contains transitions, which are pairs of states, representing the change of one state into the other. A transition may be labelled, in which case the label represents the executed action. An unlabelled transition stands for passing of time without the execution of actions (idling). The transition relation is defined by giving action rules in Structural Operational Semantics according to Plotkin ([Plo81]). These action rules are inference rules and the transition relation is the least relations satisfying the action rules.

For each operational semantics a bisimulation equivalence between states can be defined ([Par81],[Mil89]). Intuitively, two states are bisimilar if each transition from one state can be simulated by a corresponding transition from the other and vice versa.

We define a start function which provides each process term with an initial state; this is the state in which the execution of the process starts. Each operational semantics characterizes a bisimulation equivalence between process terms: two terms are bisimilar if their initial states are bisimilar.

Process Algebra aims to fit equivalences between process terms into an equational theory, i.e. an axiom system. An equational theory is said to be sound w.r.t. an equivalence if every two process terms which can be equated in the theory are equivalent. Moreover, the theory is said to be complete w.r.t. an equivalence if every two process terms which are equivalent can be equated within the theory. Since bisimulation equivalence is the only equivalence we consider in this paper, the terms sound and complete are used without further reference.

A survey of the paper

The first three sections discuss Basic Real Time Process Algebra (BPA$\rho\delta$), consisting of timed actions, sequential composition and alternative composition.

In Section 1 BPA$\rho\delta$ and its operational semantics are presented. This operational semantics results in a bisimulation equivalence where the initial state of a BPA$\rho\delta$ term has uncountably many transitions.

In Section 2 we define the theory of BPA$\rho\delta$together with a new operational semantics that abstracts from idle transitions. The action rules of this operational semantics are similar to the action rules of (untimed) ACP as given in [Gla87]. In this new operational semantics the initial state of a BPA$\rho\delta$ term has only finitely many transitions. It is designed such that it characterizes the same bisimulation equivalence as the operational semantics of the first section. However, this characterization is better suited for proving soundness and completeness of the axiom system in detail than the one of the first section.

In Section 3 completeness of BPA$\rho\delta$ is proven. We introduce the notion of a basic term. The main property of such a term is the strong correspondence between its syntax and the transitions of its initial state. Due to this correspondence we can prove completeness for basic terms. We prove as well that each term can be equated to a basic term, which induces the completeness for all terms of BPA$\rho\delta$.

In Section 4 communication operators are added, resulting in the theory ACPρ. We extend the operational semantics and prove completeness of the theory.

In Section 5 we add the important feature of integration to BPA$\rho\delta$, resulting in the theory BPA$\rho\delta I$. Integration is the alternative composition over a continuum of alternatives. Also time dependencies can be expressed: the process $\int_{v\in\langle 0,10\rangle} a(v)\cdot b(v+1)$ executes an action a in between 0 and 10 and then, one time unit after the execution of a, it executes an action b. The operational semantics of BPA$\rho\delta$ is extended to one for BPA$\rho\delta I$.

A process term in BPA$\rho\delta I$ may contain free time variables, i.e. time variables which are not bound by an integral. In order to reason with terms containing free time variables we extend the notion of a process term by allowing that it contains conditions on time variables. The definition of basic terms is extended similarly. Then we can prove in Section 6 that the theory of BPA$\rho\delta I$ is complete.

In Section 7 communication operators are added to BPA$\rho\delta I$, resulting in the theory ACPρI.

At first sight the equational theory BPA$\rho\delta I$ is undecidable, because it contains axioms that can only be validated by an uncountable number of substitutions. However, in Section 8 we give an explicit construction for reducing process terms to a normal form and in Section 9 we prove that BPA$\rho\delta I \vdash p = q$ if and only if p and q have the same normal form. Thus it is decidable for two process terms whether they are equal over this theory or not.

We do not consider recursion in this paper.

Acknowledgements. Special thanks go to Jos Baeten for his support and criticism. Jan Bergstra is thanked for giving useful comments and Jan Friso Groote, Henri Korver and Alban Ponse for discussing ideas and helping with LaTeX. Frits Vaandrager had a first idea for defining the notion of a basic term. The set-up of the last two sections was altered after a short discussion with Aart Middeldorp and Fer-Jan de Vries. Martin Wirsing is thanked for suggesting the research of completeness in ACPρ with integration.

1 An Operational Semantics

In this section we introduce the calculus of Basic Real Time Process Algebra (BPA$\rho\delta$) and provide it with an intuition by introducing the operational semantics as given in [BB91]. Let A_δ be the set of actions, containing the constant δ. The alphabet of the theory BPA$\rho\delta$ is the collection $\{a(t) \mid a \in A_\delta, t \in [0,\infty]\}$ of timed actions. In the sequel A^{time} will denote the set of timed actions without timed δ's.

The set $\mathcal{T}$ of process terms over BPA$\rho\delta$ is generated by the timed actions together with the alternative composition $+$, the sequential composition $\cdot$ and the *time shift* $\gg$. This last operator takes a nonnegative real number and a process term; $t \gg p$ denotes the part of p that starts after t. We define $\mathcal{T}$, with typical element p, inductively as follows, where $a \in A_\delta$ and $t \in [0, \infty]$:

$$p ::= a(t) \mid p + p \mid p \cdot p \mid t \gg p$$

In this section, a state is a pair consisting of a term and a point in time. The initial state of a term p is $< p, 0 >$, denoting that each process starts at time 0. We distinguish three kinds of transition relations:

$$\begin{array}{lll} \mathit{Step} & \subseteq & (\mathcal{T} \times [0,\infty]) \times A^{time} \times (\mathcal{T} \times [0,\infty]) \\ \mathit{Idle} & \subseteq & (\mathcal{T} \times [0,\infty]) \times (\mathcal{T} \times [0,\infty]) \\ \mathit{Terminate} & \subseteq & (\mathcal{T} \times [0,\infty]) \times A^{time} \times [0,\infty] \end{array}$$

They are the least relations satisfying the action rules given in this section. We write

$$\begin{array}{llll} < x,t > \xrightarrow{a(r)} < x',t' > & \text{for} & (< x,t >, a(r), < x',t' >) & \in \mathit{Step} \\ < x,t > \longrightarrow < x',t' > & \text{for} & (< x,t >, < x',t' >) & \in \mathit{Idle} \\ < x,t > \xrightarrow{a(r)} < \surd,t' > & \text{for} & (< x,t >, a(r), t') & \in \mathit{Terminate} \end{array}$$

We always have $t' = r$ in *Step* and *Terminate* and $x \equiv x'$ in *Idle*. In a text we may refer to the various kinds of transitions as:

$$\begin{array}{ll} < x,t > \xrightarrow{a(r)} < x',r > & \text{is a } a(r)\text{-transition} \\ < x,t > \longrightarrow < x,r > & \text{is an idle transition} \\ < x,t > \xrightarrow{a(r)} < \surd,r > & \text{is a terminating } a(r)\text{-transition} \end{array}$$

We may mention the *transition system* of a process term p, meaning the restriction of the transition relations to the states which can be reached from the initial state of p.

The action rules concerning alternative composition and timed actions are given in Table 1. The transitions of $p + q$ are the "union" of the transitions of p and those of q.

As an example we give the transitions of the timed action $a(1)$, denoting the process that performs action a at time 1, after which it terminates successfully. Its initial state is $< a(1), 0 >$, from where idle transitions are possible to states of the form $< a(1), t >$ with $0 < t < 1$. An idle transition is a transition which increases the time component without executing any action. In general, from a state $< a(1), t >$ an idle transition is possible to $< a(1), t' >$ whenever $t < t' < 1$. Furthermore, from each state $< a(1), t >$ with $t < 1$ an $a(1)$-transition is possible to $< \surd, 1 >$.

The transition system of the term $a(1)$ can be represented by the left-hand process diagram given in Figure 1. A process diagram is simply a pictorial representation of a transition system. It is not possible to make a picture of the transition system itself, since it has uncountably many transitions. The intuition behind such a process diagram is that the process starts at the top point. It can idle by going to a lower point without crossing any line, whereas the execution of an action a at time r is reflected by going to a dashed line at level r labelled with a. Only dashed lines may be crossed.

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

$$\frac{t < s < r}{< a(r), t > \longrightarrow < a(r), s >} \qquad \frac{< x, t > \xrightarrow{a(r)} < \surd, r >}{< x + y, t > \xrightarrow{a(r)} < \surd, r > \quad < y + x, t > \xrightarrow{a(r)} < \surd, r >}$$

$$\frac{t < s < r}{< \delta(r), t > \longrightarrow < \delta(r), s >} \qquad \frac{< x, t > \longrightarrow < x, r >}{< x + y, t > \longrightarrow < x + y, r > \quad < y + x, t > \longrightarrow < x + y, r >}$$

Table 1: Action rules for timed actions and alternative composition

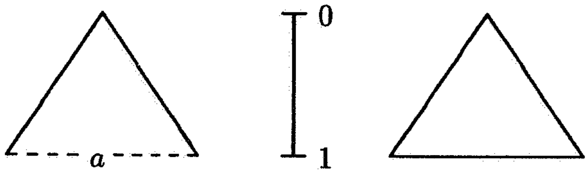

Figure 1: Process diagrams of the terms $a(1)$ and $\delta(1)$

A very particular set of atomic actions is the set of $\delta(r)$-terms. $\delta(1)$ can do nothing more then idle until 1. The initial state is $< \delta(1), 0 >$ and from each state $< \delta(1), t >$ an idle transition to $< \delta(1), t' >$ is possible whenever $t < t' < 1$; time 1 can not be reached.

The transition system of $a(1) + b(2)$ can be represented by the process diagram given in Figure 2. A state μ (in Figure 2) is of the form $< a(1) + b(2), t >$ with $0 \leq t < 1$. From μ both a terminating $a(1)$-transition to $< \surd, 1 >$ and a terminating $b(2)$-transition to $< \surd, 2 >$ are possible. However, from a state like ν of the form $< a(1) + b(2), t >$ with $1 \leq t < 2$ only a terminating $b(2)$-transition to $< \surd, 2 >$ is possible. Hence, by idling from $< a(1) + b(2), t_0 >$ to $< a(1) + b(2), t_1 >$ with $0 \leq t_0 < 1 \leq t_1 < 2$ we lose the possibility of executing the $a(1)$-summand. Thus a choice is made at time 1; if the choice is made for $b(2)$, then the summand $a(1)$ becomes redundant.

The transition system of $a(1) + \delta(1)$ consists of exactly the same relations as the transition system of $a(1)$. The summand $\delta(1)$ contributes only idle transitions that are contributed by the summand $a(1)$ as well. However, in $a(1) + \delta(2)$, the $\delta(2)$-summand contributes idle transitions which are not contributed by $a(1)$, since $\delta(2)$ has idle transitions to points in time between 1 and 2. The transition system of $a(1) + \delta(2)$ can be represented by the process diagram on the right-hand side of Figure 2.

The action rules for sequential composition and for the time shift are given in Table 2. Before t the process $t \gg p$ can only idle or do a transition to a state after t.

Finally, we give the definition of bisimulation.

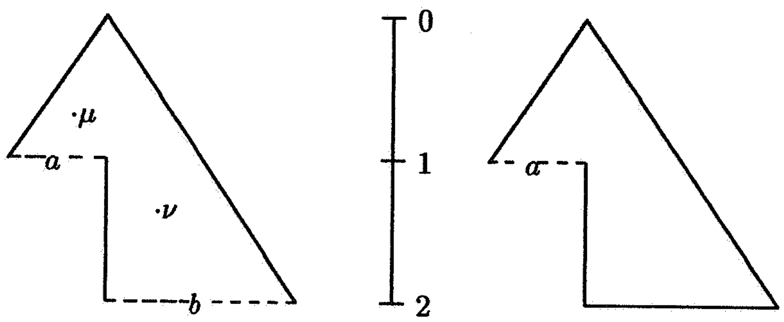

Figure 2: Process diagrams of the terms $a(1) + b(2)$ and $a(1) + \delta(2)$

$$\frac{< x,t > \xrightarrow{a(r)} < x',r >}{< x\cdot y,t > \xrightarrow{a(r)} < x'\cdot y,r >}$$

$$\frac{< x,t > \xrightarrow{a(r)} < \surd,r >}{< x\cdot y,t > \xrightarrow{a(r)} < y,r >}$$

$$\frac{< x,t > \longrightarrow < x,r >}{< x\cdot y,t > \longrightarrow < x\cdot y,r >}$$

$$\frac{t < r < s}{< s \gg x,t > \longrightarrow < s \gg x,r >}$$

$$\frac{r > s \quad < x,t > \xrightarrow{a(r)} < x',r >}{< s \gg x,t > \xrightarrow{a(r)} < x',r >}$$

$$\frac{r > s \quad < x,t > \xrightarrow{a(r)} < \surd,r >}{< s \gg x,t > \xrightarrow{a(r)} < \surd,r >}$$

$$\frac{r > s \quad < x,t > \longrightarrow < x,r >}{< s \gg x,t > \longrightarrow < x,r >}$$

Table 2: Action rules for sequential composition and time shift

Definition 1.1 *Two states* $(< p_0,t >,< q_0,t >)$ *are called* bisimilar, *denoted by* $< p_0,t > \leftrightarrow < q_0,t >$, *if there is a symmetric relation* $R \subseteq (\mathcal{T} \times [0,\infty]) \times (\mathcal{T} \times [0,\infty])$ *such that:*

1. $< p_0,t >$ *and* $< q_0,t >$ *are related by* R.

2. *If* $< p,s > \xrightarrow{a(r)} < p',r >$ *and* $R(< p,s >,< q,s >)$, *then there is a transition* $< q,s > \xrightarrow{a(r)} < q',r >$ *such that* $R(< p',r >,< q',r >)$.

3. *If* $< p,s > \longrightarrow < p,r >$ *and* $R(< p,s >,< q,s >)$, *then* $< q,s > \longrightarrow < q,r >$

4. *If* $< p,s > \xrightarrow{a(r)} \surd$ *and* $R(< p,s >,< q,s >)$, *then there is a transition* $< q,s > \xrightarrow{a(r)} \surd$.

2 Basic Real Time Process Algebra

2.1 The theory of BPAρ

BPA$\rho\delta$ is the theory of Basic Real Time Process Algebra ([BB91]). It consists of the standard axioms A1-5 of Basic Process Algebra, extended with some axioms describing real-time properties and defining the time shift. The axioms of BPA$\rho\delta$ are given in Table 3. Let $a \in A_\delta$. We abbreviate $\delta(0)$ to δ. The laws ATA3,4 are generalizations of the BPAδ laws A6 and A7.

A1		$X + Y = Y + X$
A2		$(X + Y) + Z = X + (Y + Z)$
A3		$X + X = X$
A4		$(X + Y) \cdot Z = X \cdot Z + Y \cdot Z$
A5		$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$
ATA1a		$a(0) = \delta$
ATA1b		$a(\infty) = \delta(\infty)$
ATA2	$t \leq s$	$\delta(t) + \delta(s) = \delta(s)$
ATA3		$\delta(t) \cdot X = \delta(t)$
ATA4		$a(t) + \delta(t) = a(t)$
SH1		$a(t) \cdot X = a(t) \cdot (t \gg X)$
SH2a	$t < s$	$t \gg a(s) = a(s)$
SH2b	$t \geq s$	$t \gg a(s) = \delta(t)$
SH3		$t \gg (X + Y) = (t \gg X) + (t \gg Y)$
SH4		$t \gg (X \cdot Y) = (t \gg X) \cdot Y$

Table 3: An axiom system for BPA$\rho\delta$

Using BPA$\rho\delta$ we can prove:

$$\begin{aligned} 5 \gg (a(4) + b(6) + c(7) \cdot d(8)) &= b(6) + c(7) \cdot d(8) \\ 5 \gg (a(4) + b(3)) &= \delta(5) \\ \delta(1) + a(2) \cdot b(3) + \delta(3) \cdot c(4) &= a(2) \cdot b(3) + \delta(3) \end{aligned}$$

2.2 Some definitions

For a process term p we now introduce its *ultimate delay* $U(p)$, which is the first moment in time that p can not reach by idling only. The ultimate delay has been introduced in [BB91], where it was defined for terms not containing time shifts and extended to $\mathcal{T}$ by putting $p = q \Longrightarrow U(p) = U(q)$. Here the ultimate delay is defined on the syntax only, so it can be used in the operational semantics. In [MT90] Moller & Tofts have introduced a similar construct, which they called the maximum delay.

The ultimate delay is defined inductively as follows, where $a \in A_\delta$.

$$\begin{array}{lcl} U(a(t)) & = & t \\ U(X+Y) & = & max\{U(X), U(Y)\} \\ U(X \cdot Y) & = & U(X) \\ U(t \gg X) & = & max\{U(X), t\} \end{array}$$

The *size* of a term is the number of operators in the term. And the *depth* of a term is the longest chain of sequential compositions in the term. It is defined inductively as follows, where $a \in A_\delta$.

$$\begin{array}{lcl} depth(a(t)) & = & 1 \\ depth(X+Y) & = & max\{depth(X), depth(Y)\} \\ depth(X \cdot Y) & = & depth(X) + depth(Y) \\ depth(t \gg X) & = & depth(X) \end{array}$$

Note that depth can not be defined on the theory; for example, $\delta(1) \cdot a(2) = \delta(1)$, but the two terms have different depths.

Two terms p, q are said to be syntactically equivalent, denoted by $p \equiv q$, if they are constructed in exactly the same way from the atomic actions and the constructors. Normally we are not much interested whether two terms are syntactically equivalent, but more whether they are equal modulo the axioms A1 and A2. This is denoted by $p \cong q$ and we say that p and q have the same form. If there is a derivation in the theory BPA$\rho\delta$ connecting two terms p and q, then this is denoted by $p = q$. Moreover, we have two notions of summand inclusion. A term p is said to be a *syntactic* summand of q, denoted by $p \sqsubseteq q$, if $p \cong q$ or there is a q' such that $p \cong q + q'$. And p is called a *derivable* summand of q, denoted by $p \subseteq q$, if $p + q = q$.

2.3 An operational semantics for BPAρ with finite transition systems

In the first section the operational semantics has been presented according to [BB91]. There each state was a pair of an expression and a time stamp; an idle transition increased the time while the expression remained the same. We now define an operational semantics without idle transitions, which characterizes the same bisimulation equivalence as the operationals semantics of the previous section. The transition system of $a(1)$ will contain only one labelled arrow. This new operational semantics is analogous to the operational semantics for ACP presented in [Gla87], where the following transitions occur:

$$a \xrightarrow{a} \surd \quad \text{and} \quad a \cdot p \xrightarrow{a} p$$

In a real-time setting we have to take the time stamps into account. Consider the term $a(t) \cdot p$. After executing the $a(t)$-action only that part of p can be executed which starts after t. Now we have the following transitions:

$$a(t) \xrightarrow{a(t)} \surd \quad \text{and} \quad a(t) \cdot p \xrightarrow{a(t)} t \gg p$$

In Figure 3 the transition systems for the terms $a(1)$ and $a(1) + b(2)$ are given, together with the corresponding process diagrams.

In Table 4 the action rules of this operational semantics are given. Now every state is a term from $\mathcal{T}$ and every transition is labelled by a timed atomic action. There are no idle

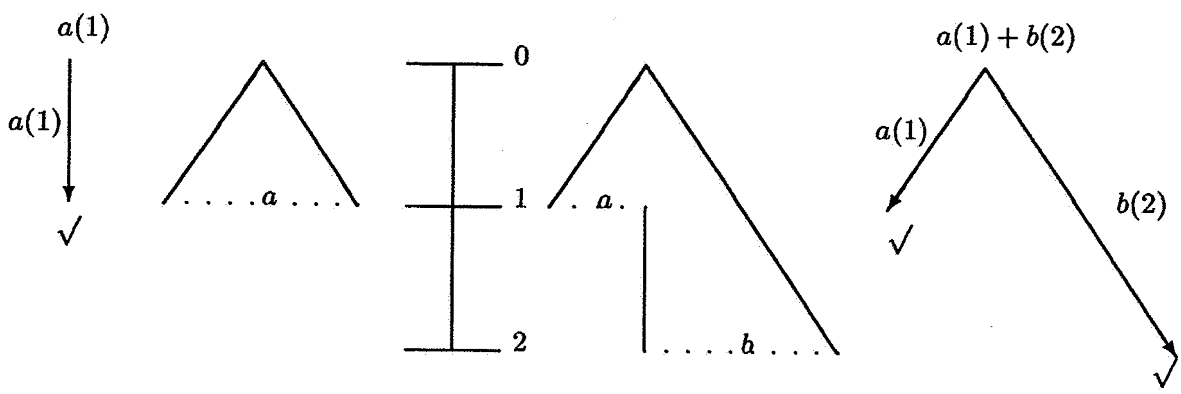

Figure 3: Process diagrams and transition systems for the terms $a(1)$ and $a(1) + b(2)$

transitions and every term is its own initial state. The new operational semantics concerns three transition relations:

$$\begin{array}{lll} Step & \subseteq & \mathcal{T} \times A^{time} \times \mathcal{T} \\ Terminate & \subseteq & \mathcal{T} \times A^{time} \\ Deadlock & \subseteq & \mathcal{T} \times [0, \infty] \end{array}$$

We write:

$$\begin{array}{llll} p \xrightarrow{a(t)} p' & \text{for} & (p, a(t), p') \in Step \\ p \xrightarrow{a(t)} \surd & \text{for} & (p, a(t)) \in Terminate \\ p \xrightarrow{\delta(t)} \surd & \text{for} & (p, t) \in Deadlock \end{array}$$

The transition relations *Step* and *Terminate* are defined as the least relations satisfying the action rules of the top part of Table 4. The transition relation *Deadlock* is introduced in order to distinguish by bisimulation process terms that only differ in their deadlock behaviour. Note that the transition relation *Idle* did that job in the previous section. *Deadlock* contains all pairs (p, t) for which p has an initial deadlock at time t. A process term p has a possible initial deadlock if and only if it can idle beyond its latest initial action time. Therefore, in order to define the relation *Deadlock* properly, we now introduce $L(p)$. This is the maximum of points in time at which p can execute an initial action. The process term p can do a δ-transition at $U(p)$ if and only if $U(p) > L(p)$.

$$\begin{array}{lll} L(\delta(t)) & = & 0 \\ L(a(t)) & = & t \\ L(X + Y) & = & max\{L(X), L(Y)\} \\ L(X \cdot Y) & = & L(X) \\ L(r \gg X) & = & \begin{cases} L(X) & \text{if } L(X) > r \\ 0 & \text{otherwise} \end{cases} \end{array}$$

We adapt the definition of bisimulation.

Definition 2.1 *A pair* $p_0, q_0 \in \mathcal{T}$ *is called* bisimilar, *denoted by* $p_0 \leftrightarrow q_0$, *if there is a symmetric relation* $R \subseteq \mathcal{T} \times \mathcal{T}$, *called a bisimulation relation, such that:*

$a \in A,\ t \in \langle 0, \infty \rangle$			
atom	:	$a(t) \xrightarrow{a(t)} \surd$	
$\cdot$	:	$\dfrac{p \xrightarrow{a(t)} \surd}{p \cdot q \xrightarrow{a(t)} t \gg q}$	$\dfrac{p \xrightarrow{a(t)} p'}{p \cdot q \xrightarrow{a(t)} p' \cdot q}$
$+$	:	$\dfrac{p \xrightarrow{a(t)} \surd}{p+q \xrightarrow{a(t)} \surd, q+p \xrightarrow{a(t)} \surd}$	$\dfrac{p \xrightarrow{a(t)} p'}{p+q \xrightarrow{a(t)} p', q+p \xrightarrow{a(t)} p'}$
$\gg$	:	$\dfrac{s < t \quad p \xrightarrow{a(t)} \surd}{s \gg p \xrightarrow{a(t)} \surd}$	$\dfrac{s < t \quad p \xrightarrow{a(t)} p'}{s \gg p \xrightarrow{a(t)} p'}$
δ	:	$\dfrac{U(p) > L(p)}{p \xrightarrow{\delta(U(p))} \surd}$	

Table 4: Action rules for BPA$\rho\delta$

1. p_0 and q_0 are related by R.

2. If $p \xrightarrow{a(r)} p'$ and $R(p,q)$, then there is a transition $q \xrightarrow{a(r)} q'$ such that $R(p',q')$.

3. If $p \xrightarrow{a(r)} \surd$ and $R(p,q)$, then there is a transition $q \xrightarrow{a(r)} \surd$.

In [Klu91] it is proven that bisimulation equivalence $\leftrightarrows$ is a *congruence*. This means that if $p \leftrightarrows p'$ and $q \leftrightarrows q'$, then $p+q \leftrightarrows p'+q'$ and $p \cdot q \leftrightarrows p' \cdot q'$ and $t \gg p \leftrightarrows t \gg p'$. Groote has introduced a special format for action rules in [Gro89], the so-called $ntyft/ntyxt$-format, and proved that a bisimulation equivalence is always a congruence if it is characterized by an operational semantics with action rules in this format. In [Klu91] the bisimulation equivalence of this paper is defined by a transition relation with action rules in Groote's format.

The following theorem says that the theory of BPA$\rho\delta$ is *sound*. This means that if BPA$\rho\delta \vdash p = q$ then $p \leftrightarrows q$. Since the congruence of $\leftrightarrows$ is already guaranteed, it can be proven by showing for each axiom separately that the process terms on the left- and on the right-hand side are bisimilar (for an arbitrary instantiation).

Theorem 2.2 *The theory of* BPA$\rho\delta$ *is sound.*

In this section we have characterized bisimulation equivalence by a different operational semantics than in the previous section. In [Klu91] it is proven that these two different operational semantics characterize the same bisimulation equivalence.

3 Completeness

In this section we prove that BPA$\rho\delta$ is *complete*. This means that if $p \leftrightarrow q$, then BPA$\rho\delta \vdash p = q$. First we show that each process term can be reduced to a *basic* form.

3.1 Basic terms

A basic term will be a term that does not contain "redundant" parts. For example, the terms $a(5) \cdot b(6) + \delta(1)$ and $a(5) \cdot (b(6) + c(4))$ contain parts that can be removed by application of the axioms of BPA$\rho\delta$; they are derivably equal to the term $a(5) \cdot b(6)$, which does not contain any redundant parts. We will be able to prove completeness for basic terms, since the transitions of a basic term corresponds directly with its syntax. By showing that each process term is equal to a basic term, completeness follows for general terms.

Basic terms have ascending time stamps

For the term $a(t) \cdot p$ to be a basic term, it is required that p is a basic term that either starts after t or is equal to δ. So if $\mathcal{B}$ denotes the set of basic terms, then

$$\begin{array}{llll} a(1) \cdot b(2) & \in \mathcal{B} & a(1) \cdot (b(2) + c(1)) & \notin \mathcal{B} \\ a(1) \cdot \delta & \in \mathcal{B} & a(1) \cdot \delta(1) & \notin \mathcal{B} \end{array}$$

Thus, while constructing $\mathcal{B}$ we have to keep track of the start time of each basic term. Therefore we will construct for $t \in [0, \infty)$ a set $\mathcal{B}(t)$, containing the basic terms starting after t (together with δ). Then $a(t) \cdot p$ is a basic term iff $p \in \mathcal{B}(t)$. Furthermore, $\mathcal{B}$ will be equal to $\mathcal{B}(0)$. (Note that if $t \leq t'$, then $\mathcal{B}(t) \supseteq \mathcal{B}(t')$).

Basic terms do not contain redundant deadlocks

Moreover, a basic term will not contain timed deadlocks that do not contribute to its deadlock behaviour. In other words, if $p + \delta(t)$ is a basic term then it can do a $\delta(t)$-transition to $\surd$. For example:

$$\begin{array}{llll} a(1) + \delta(2) & \in \mathcal{B} & a(1) + \delta(2) + \delta(1) & \notin \mathcal{B} \\ \delta(3) + \delta(3) & \in \mathcal{B} & a(3) + \delta(3) & \notin \mathcal{B} \end{array}$$

Thus, while constructing $\mathcal{B}$ we have to keep track of the possible initial deadlocks of each basic term. Therefore we will construct for $s \in [0, \infty]$ and $t \in [0, \infty)$ a set $\mathcal{B}^s(t)$ containing the basic terms starting after t with an initial deadlock at s. If a basic term p (starting after t) has no initial deadlock, then we define $p \in \mathcal{B}^\infty(t)$. The sum of two basic terms p, q is again a basic term if either their initial deadlocks coincide or if p has an initial deadlock at s and q does not have an initial deadlock and $s > U(q)$. The set $\mathcal{B}(t)$ is defined to be the union over $s > t$ of the sets $\mathcal{B}^s(t)$. (Note that if $s \leq t$, then $\mathcal{B}^s(t) = \emptyset$).

Definition 3.1 *Let $p, q \in \mathcal{T}$, $a \in A$, $r \in \langle 0, \infty \rangle$, $s \in [0, \infty]$ and $t \in [0, \infty\rangle$.*

$\mathcal{B}, \mathcal{B}(t)$ and $\mathcal{B}^s(t)$ are the smallest sets satisfying:

$$\begin{array}{rlll}
 & \delta & \in & \mathcal{B}^\infty(t) \\
t < r & a(r) & \in & \mathcal{B}^\infty(t) \\
t < s & \delta(s) & \in & \mathcal{B}^s(t) \\
t < r \ \wedge \ p \in \mathcal{B}(r) & a(r) \cdot p & \in & \mathcal{B}^\infty(t) \\
p \in \mathcal{B}^s(t) \ \wedge \ q \in \mathcal{B}^s(t) & p + q & \in & \mathcal{B}^s(t) \\
p \in \mathcal{B}^s(t) \ \wedge \ q \in \mathcal{B}^\infty(t) \ \wedge \ s > U(q) & p + q, q + p & \in & \mathcal{B}^s(t) \\
 & & & \\
 & \mathcal{B}(t) & = & \cup_s \mathcal{B}^s(t) \\
 & \mathcal{B} & = & \mathcal{B}(0)
\end{array}$$

In a basic term only prefixed multiplication is used and the time shift $\gg$ does not occur. If p is a basic term, then we have allowed $p + \delta$ to be a basic term. For convenience of notation we extend the definition of $\cong$ (equality modulo A1,2) by putting $p \cong p + \delta$. Then, using the convention $\sum_{i \in \emptyset} p_i \cong \delta$, every basic term is of the form

$$\sum_i a_i(t_i) \cdot p_i \ + \ \sum_j b_j(s_j)$$

where $p_i \in \mathcal{B}(t_i)$ and $a_i \in A$ and $b_j \in A_\delta$.

Theorem 3.2 *For each term $p \in \mathcal{T}$ there is a basic term p_b such that* BPA$\rho\delta \vdash p = p_b$.

Proof. By induction on the size of p.

1. $p \equiv a(t)$
   If $t \in \langle 0, \infty \rangle$, then $a(t) \in \mathcal{B}$. And if $t = 0$ or $t = \infty$, then $a(t)$ is equal to the basic term δ resp. $\delta(\infty)$.

2. $p \equiv q \cdot q'$
   By induction there are $q_b, q'_b \in \mathcal{B}$ such that BPA$\rho\delta \vdash q = q_b$ and BPA$\rho\delta \vdash q' = q'_b$. Note that the sizes of q_b and q'_b may be greater than the sizes of q and q'. We now prove by induction to size that for all $q_b, q'_b \in \mathcal{B}$ there is a $z \in \mathcal{B}$ with $z = q_b \cdot q'_b$. Assume

   $$\begin{array}{rcl} q_b & \cong & \sum_{i \in I} a_i(r_i) \cdot q_i + \sum_{j \in J} b_j(s_j) \\ q'_b & \cong & \sum_{k \in K} c_k(t_k) \cdot q'_k + \sum_{l \in L} d_l(u_l) \end{array}$$

   Construct $\overline{q_b}$ from q_b by removing the δ-summands:

   $$\begin{array}{rcl} \overline{J} & = & \{j \in J \mid b_j \neq \delta\} \\ \overline{q_b} & \cong & \sum_{i \in I} a_i(r_i) \cdot q_i + \sum_{j \in \overline{J}} b_j(s_j) \end{array}$$

   By induction there is for each $i \in I$ a $z_i \in \mathcal{B}$ such that $z_i = q_i \cdot q'_b$. Since $q_b \in \mathcal{B}$, it follows that $q_i \in \mathcal{B}(r_i)$ for all i. Then clearly $z_i \in \mathcal{B}(r_i)$ for all i. Now define for each r

   $$\begin{array}{rcl} K^r & = & \{k \in K | t_k > r\} \\ L^r & = & \{l \in L | u_l > r\} \end{array}$$

By constructing $\overline{z}$ as follows we have $\overline{z} = \overline{q_b} \cdot q_b'$.

$$\overline{z} \cong \sum_{i\in I} a_i(r_i) \cdot z_i + \sum_{j\in \overline{J}} b_j(s_j) \cdot \{ \sum_{k\in K^{s_j}} c_k(t_k) \cdot q_k' + \sum_{l\in L^{s_j}} d_l(u_l)\}$$

Finally we can construct $z \in \mathcal{B}$ satisfying $z = q_b \cdot q_b'$ by taking

$$z \equiv \begin{cases} \overline{z} + \delta(U(q_b)) & \text{if } U(q_b) > U(\overline{q_b}) \\ \overline{z} & \text{otherwise} \end{cases}$$

3. $p \equiv q + q'$
By induction there are $q_b, q_b' \in \mathcal{B}$ such that BPA$\rho\delta \vdash q = q_b$ and BPA$\rho\delta \vdash q' = q_b'$. Construct $\overline{q_b}$ and $\overline{q_b}'$ from q_b and q_b' by removing the δ-summands. Then

$$p = \begin{cases} \overline{q_b} + \overline{q_b}' + \delta(U(q+q')) \in \mathcal{B} & \text{if } U(q+q') > U(\overline{q_b} + \overline{q_b}') \\ \overline{q_b} + \overline{q_b}' \in \mathcal{B} & \text{otherwise} \end{cases}$$

4. $p \equiv t \gg q$
By induction there is a $q_b \in \mathcal{B}$ such that BPA$\rho\delta \vdash q = q_b$, say

$$q_b \cong \textstyle\sum_{i\in I} a_i(r_i) \cdot q_i + \sum_{j\in J} b_j(s_j)$$

Define $I^r = \{i \in I | r_i > r\}$ and $J^r = \{j \in J | s_j > r\}$. Then

$$p = \begin{cases} \delta(t) \in \mathcal{B} & \text{if } I^t \cup J^t = \emptyset \\ \sum_{i\in I^t} a_i(r_i) \cdot q_i + \sum_{j\in J^t} b_j(s_j) \in \mathcal{B} & \text{otherwise} \end{cases}$$

□

For $p \in \mathcal{B}$ and $a \in A_\delta$ and $t \in [0, \infty]$ we have

$$\begin{array}{rcl} p \xrightarrow{a(t)} \surd & \iff & a(t) \sqsubseteq p \\ p \xrightarrow{a(t)} t \gg p' & \iff & a(t) \cdot p' \sqsubseteq p \end{array}$$

The following lemma states that basic terms have ascending time stamps.

Lemma 3.3 $\forall p \in \mathcal{B}$ $\qquad p \xrightarrow{a(t)} t \gg p' \implies t \gg p' \leftrightarrow p'$

Proof. By definition of basic terms and $a(t) \cdot p' \sqsubseteq p$ we know $p' \in \mathcal{B}(t)$. Hence, every transition of p' has a time stamp greater than t. And thus if $p' \xrightarrow{b(s)} p''$, then $t \gg p' \xrightarrow{b(s)} p''$ as well. Similarly for terminating transitions of p'. □

3.2 Completeness of BPA$\rho\delta$

We can now prove that the theory BPA$\rho\delta$ is complete.

Theorem 3.4 $\forall p, q \in \mathcal{T}$ $\qquad p \leftrightarrow q \implies$ BPA$\rho\delta \vdash p = q$

Proof. Lemma 3.2 together with soundness implies that it is sufficient to consider basic terms only. We will prove by induction on the depth of p that $p \subseteq q$. By symmetry $q \subseteq p$ follows and we are done. Assume

$$p \cong \sum_i a_i(t_i) \cdot p_i + \sum_j b_j(s_j)$$

Since p is a basic term, there is for each j a transition $p \xrightarrow{b_j(s_j)} \surd$. We assume $p \leftrightarrows q$, so there is for each j a transition $q \xrightarrow{b_j(s_j)} \surd$ and thus $b_j(s_j) \sqsubseteq q$.

Similarly there is for each i a term q' such that $q \xrightarrow{a_i(t_i)} t_i \gg q'$ and $t_i \gg p_i \leftrightarrows t_i \gg q'$. Since p and q are basic terms, Lemma 3.3 implies

$$p_i \leftrightarrows t_i \gg p_i \leftrightarrows t_i \gg q' \leftrightarrows q'$$

and thus by induction it follows that $p_i = q'$. Together with $a_i(t_i) \cdot q' \sqsubseteq q$ we may conclude that $a_i(t_i) \cdot p_i \subseteq q$. □

4 Parallelism and Synchronization

4.1 Introduction

Now we are ready to introduce parallelism and synchronization, resulting in the theory ACPρ from [BB91]. We will use as much as possible from ACP (without time) and we shall discuss only those cases in which we have to take the time information into account. One of the operators is the left merge, which is an auxiliary operator that allows us to define the parallel merge operator $\|$ in finitely many axioms. In standard ACP (without time) the term $(a \cdot X) \mathbin{⫗} Y$ denotes the process in which the left component $a \cdot X$ executes his first action a, resulting in $X \| Y$. In the real-time setting it is a bit more subtle. Consider for example the process $(a(t) \cdot X) \mathbin{⫗} Y$. By executing the a action at time t the whole process must proceed in time. Whenever Y can wait till after t (so if $t < U(Y)$), then $(a(t) \cdot X) \mathbin{⫗} Y$ can execute action a at time t, resulting in $(t \gg X) \| (t \gg Y)$. Otherwise (if $t \geq U(Y)$) a deadlock at time $U(Y)$ occurs. So

$$\begin{aligned} a(2) \mathbin{⫗} b(3) &= a(2) \cdot b(3) \\ b(3) \mathbin{⫗} a(2) &= \delta(2) \end{aligned}$$

In the first example the right component $b(3)$ can wait until the left component $a(2)$ executes its first action. In the second example however, we see that the right component $a(2)$ cannot wait long enough and a deadlock is the result.

We now extend the notion of a process term. The set $\mathcal{T}$, with typical element p, is defined inductively as follows, where $a \in A_\delta$, $t \in [0, \infty]$ and $H \subseteq A$:

$$p ::= a(t) \mid p + p \mid p \cdot p \mid t \gg p \mid p \| p \mid p \mathbin{⫗} p \mid p | p \mid \partial_H(p)$$

The definition of the ultimate delay is extended as follows.

$$\begin{aligned} U(X \square Y) &= min\{U(X), U(Y)\} \qquad \square \in \{\|, \mathbin{⫗}, |\} \\ U(\partial_H(X)) &= U(X) \end{aligned}$$

We assume a communication function $| : A_\delta \times A_\delta \longrightarrow A_\delta$ which is commutative and associative and has δ as zero element. Communication between atomic actions happening at different times is not possible. Thus if $a|b = c$ then

$$\begin{aligned} a(2)|b(2) &= c(2) \\ a(1)|b(3) &= \delta(1) \end{aligned}$$

For $H \subseteq A$ the operator ∂_H is defined on A_δ as follows:

$$\begin{array}{ll} a \notin H & \partial_H(a) = a \\ a \in H & \partial_H(a) = \delta \end{array}$$

4.2 The theory of ACPρ

The axiom system for ACPρ consists of BPA$\rho\delta$ together with the axioms of Table 5. The names of the axioms have been taken from untimed ACP. If an axiom is derived from an axiom of untimed ACP but not totally similar, then this is reflected by adding AT (Absolute Time) to its name.

M1		$X \| Y = X \lfloor\!\lfloor Y + Y \lfloor\!\lfloor X + X \mid Y$
LM1a	$t < U(Y)$	$a(t) \lfloor\!\lfloor Y = a(t) \cdot Y$
LM1b	$t \geq U(Y)$	$a(t) \lfloor\!\lfloor Y = \delta(U(Y))$
LM2a	$t < U(Y)$	$(a(t) \cdot X) \lfloor\!\lfloor Y = a(t) \cdot (X \| Y)$
LM2b	$t \geq U(Y)$	$(a(t) \cdot X) \lfloor\!\lfloor Y = \delta(U(Y))$
LM3		$(X_1 + X_2) \lfloor\!\lfloor Y = X_1 \lfloor\!\lfloor Y + X_2 \lfloor\!\lfloor Y$
CM1a		$a(t) \mid b(t) = (a \mid b)(t)$
CM1b	$t \neq s$	$a(t) \mid b(s) = \delta(min\{t, s\})$
CM2		$(a(t) \cdot X) \mid b(s) = (a(t) \mid b(s)) \cdot X$
CM3		$a(t) \mid (b(s) \cdot Y) = (a(t) \mid b(s)) \cdot Y$
CM4		$(a(t) \cdot X) \mid (b(s) \cdot Y) = (a(t) \mid b(s)) \cdot (X \| Y)$
CM5		$(X_1 + X_2) \mid Y = X_1 \mid Y + X_2 \mid Y$
CM6		$X \mid (Y_1 + Y_2) = X \mid Y_1 + X \mid Y_2$
ATD1		$\partial_H(a(t)) = \partial_H(a)(t)$
D3		$\partial_H(X + Y) = \partial_H(X) + \partial_H(Y)$
D4		$\partial_H(X \cdot Y) = \partial_H(X) \cdot \partial_H(Y)$

Table 5: An axiom system for ACPρ

The axiom CM1 is exactly the same as in ACP. However, together with the axioms for the left merge it does not result in arbitrary interleaving, since the time stamps of the atomic actions determine the possible orderings. For example

$$\begin{aligned} a(2)\|b(3) &= a(2)\mathbin{\|\!\_}\,b(3) + b(3)\mathbin{\|\!\_}\,a(2) + a(2)|b(3) \\ &= a(2)\cdot b(3) + \delta(2) + \delta(2) \\ &= a(2)\cdot b(3) \end{aligned}$$

4.3 An operational semantics for ACPρ

Table 6 contains the action rules for the new operators $\|$, $|$, $\mathbin{\|\!\_}$ and ∂_H. The action rules for the operators of BPA$\rho\delta$ can be found in Table 4. Remember that if the process $a(t)\cdot p$ executes the $a(t)$-action, it evolves to $t \gg p$. If $a(t)\cdot p$ executes this $a(t)$ in a parallel composition with q, the "increase in time" must hold for q as well. In other words, $(a(t)\cdot p)\|q$ evolves to $(t \gg p)\|(t \gg q)$ by executing a at time t whenever q can wait until t. The check whether q can increase its time, is denoted by $t < U(q)$.

In order to let the deadlock transition defined in Table 4 be valid for ACPρ, we need to extend the definition of $L(p)$ to terms that contain communication and deadlock operators.

$$\begin{aligned}
initact(\delta(t)) &= \emptyset \\
initact(a(t)) &= \begin{cases} \emptyset & \text{if } t \in \{0,\infty\} \\ \{a(t)\} & \text{otherwise} \end{cases} \\
initact(X+Y) &= initact(X) \cup initact(Y) \\
initact(X\cdot Y) &= initact(X) \\
initact(s \gg X) &= \{a(t) \in initact(X) \mid t > s\} \\
initact(X\|Y) &= initact(X\mathbin{\|\!_}\,Y + Y\mathbin{\|\!_}\,X + X|Y) \\
initact(X\mathbin{\|\!_}\,Y) &= \{a(t) \in initact(X) \mid t < U(Y)\} \\
initact(X|Y) &= \{c(t) \mid \exists a,b \quad a(t) \in initact(X) \ \wedge\ b(t) \in initact(Y) \ \wedge\ a|b = c \neq \delta\} \\
initact(\partial_H(X)) &= \{a(t) \mid a(t) \in initact(X) \ \wedge\ a \notin H\} \\
\\
L(X) &= max\{t \mid \exists a \quad a(t) \in initact(X)\}
\end{aligned}$$

In [Klu91] it is proven that bisimulation equivalence is a congruence for the added operators as well. Again, action rules in Groote's format can be given for these operators. The following theorem can be proven by checking it for each axiom separately.

Theorem 4.1 *The theory of* ACPρ *is sound.*

In [Klu91] it is proven that for ACPρ the operational semantics of this section characterizes the same bisimulation equivalence as the operational semantics of [BB91].

4.4 Basic terms

We now prove that every process term has a basic form. First we need a lemma.

Lemma 4.2 *For each* $p, q \in \mathcal{B}$ *there is a* $z \in \mathcal{B}$ *such that* ACP$\rho \vdash z = p\|q$

Proof. We use induction on size. Assume

$$\begin{aligned} p &\cong \textstyle\sum_{i\in I} a_i(r_i)\cdot p_i + \sum_{j\in J} b_j(s_j) \\ q &\cong \textstyle\sum_{k\in K} c_k(t_k)\cdot q_k + \sum_{l\in L} d_l(u_l) \end{aligned}$$

$a, b \in A$

$$\|, \lfloor\!\lfloor \quad : \quad \frac{p \xrightarrow{a(t)} p' \quad t < U(q)}{p\|q \xrightarrow{a(t)} p'\|(t \gg q)\,, \quad q\|p \xrightarrow{a(t)} (t \gg q)\|p'\,, \quad p\lfloor\!\lfloor q \xrightarrow{a(t)} p'\|(t \gg q)}$$

$$\frac{p \xrightarrow{a(t)} \surd \quad t < U(q)}{p\|q \xrightarrow{a(t)} t \gg q\,, \quad q\|p \xrightarrow{a(t)} t \gg q\,, \quad p\lfloor\!\lfloor q \xrightarrow{a(t)} t \gg q}$$

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

$$\|, | \quad : \quad \frac{p \xrightarrow{a(t)} p' \quad q \xrightarrow{b(t)} q'}{p\|q \xrightarrow{c(t)} p'\|q'\,, \quad p|q \xrightarrow{c(t)} p'\|q'} \qquad \frac{p \xrightarrow{a(t)} \surd \quad q \xrightarrow{b(t)} \surd}{p\|q \xrightarrow{c(t)} \surd\,, \quad p|q \xrightarrow{c(t)} \surd}$$

$$\frac{p \xrightarrow{a(t)} \surd \quad q \xrightarrow{b(t)} q'}{p\|q \xrightarrow{c(t)} q'\,, \quad q\|p \xrightarrow{c(t)} q'\,, \quad p|q \xrightarrow{c(t)} q'\,, \quad q|p \xrightarrow{c(t)} q'}$$

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

Table 6: Additional action rules for ACPρ

We define

$$\begin{aligned} \bar{I} &= \{i \in I \mid r_i < U(q)\} \\ \bar{J} &= \{j \in J \mid s_j < U(q)\} \\ \bar{K} &= \{k \in K \mid t_k < U(p)\} \\ \bar{L} &= \{l \in L \mid u_l < U(p)\} \end{aligned}$$

Then $p\|q$ is equal to the following process term. The term $\delta(U(p\|q))$ is added for the case that $\bar{I} \cup \bar{J} \cup \bar{K} \cup \bar{L} = \emptyset$.

$$\begin{aligned} & \textstyle\sum_{i \in \bar{I}} a_i(r_i) \cdot (p_i\|q) + \sum_{j \in \bar{J}} b_j(s_j) \cdot q \\ + \; & \textstyle\sum_{k \in \bar{K}} c_k(t_k) \cdot (p\|q_k) + \sum_{l \in \bar{L}} d_l(u_l) \cdot p \\ + \; & \textstyle\sum_{(i,k) \in I \times K} (a_i(r_i)|c_k(t_k)) \cdot (p_i\|q_k) + \sum_{(i,l) \in I \times L} (a_i(r_i)|d_l(u_l)) \cdot p_i \\ + \; & \textstyle\sum_{(j,k) \in J \times K} (b_j(s_j)|c_k(t_k)) \cdot q_k + \sum_{(j,l) \in J \times L} b_j(s_j)|d_l(u_l) \\ + \; & \delta(U(p\|q)) \end{aligned}$$

An expression $a(t)|b(t')$ is equal to $c(s)$ for some $c \in A_\delta$. And according to the induction hypothesis the terms $p_i\|q$ and $p\|q_k$ and $p_i\|q_k$ all have basic forms. Thus $p\|q$ is equal to a process term that does not contain $\|$, $\lfloor\!\lfloor$, $|$ and ∂_H. We know already that such a term has a basic form. □

Theorem 4.3 *For each term $p \in \mathcal{T}$ there is a basic term p_b such that* ACP$\rho \vdash p = p_b$

Proof. The proof uses induction on the size of p. We discuss four cases; the other cases can be proven as in Lemma 3.2.

1. $p \equiv q|q'$. By induction there are $q_b, q'_b \in \mathcal{B}$ with ACP$\rho \vdash q = q_b$ and ACP$\rho \vdash q' = q'_b$. Let
$$\begin{array}{rcl} q_b & \cong & \sum_i a_i(r_i) \cdot q_i + \sum_j b_j(s_j) \\ q'_b & \cong & \sum_k c_k(t_k) \cdot q'_k + \sum_l d_l(u_l) \end{array}$$
After applying CM1-6 sufficiently many times we get
$$\begin{array}{rcl} p & = & \sum_{(i,k)}(a_i(r_i)|c_k(t_k)) \cdot (q_i \| q'_k) \;+\; \sum_{(i,l)}(a_i(r_i)|d_l(u_l)) \cdot q_i \;+ \\ & & \sum_{(j,k)}(b_j(s_j)|c_k(t_k)) \cdot q'_k \;+\; \sum_{(j,l)} b_j(s_j)|d_l(u_l) \end{array}$$
Each $a(t)|b(t')$ reduces to some $c(s)$. Furthermore, according to Lemma 4.2 the terms $q_i \| q'_k$ all have a basic form. Thus p is equal to a process term not containing $\|$, $\mathbin{\|\!\_}$, $|$ and ∂_H. And we know already that such a term has a basic form.

2. $p \equiv q \mathbin{\|\!\_} q'$. By induction there are $q_b, q'_b \in \mathcal{B}$ with ACP$\rho \vdash q = q_b$ and ACP$\rho \vdash q' = q'_b$. Let
$$q_b \cong \sum_{i \in I} a_i(r_i) \cdot q_i + \sum_{j \in J} b_j(s_j)$$
Define $\bar{I} = \{i \in I \mid r_i < U(q')\}$ and $\bar{J} = \{j \in J \mid s_j < U(q')\}$. After applying LM1-3 sufficiently many times we get
$$p = \sum_{i \in \bar{I}} a_i(r_i) \cdot (q_i \| q'_b) + \sum_{j \in \bar{J}} b_j(s_j) \cdot q'_b + \delta(U(q'))$$
The term $\delta(U(q'))$ is added for the case that $\bar{I} \cup \bar{J} = \emptyset$. By Lemma 4.2 $q_i \| q'_b$ has a basic form. Then p is equal to a term that does not contain $\|$, $\mathbin{\|\!\_}$, $|$ and ∂_H. And such a term has a basic form.

3. $p \equiv q \| r$. This follows from Lemma 4.2 and the induction hypothesis.

4. $p \equiv \partial_H(q)$. By induction there is a $q_b \in \mathcal{B}$ such that ACP$\rho \vdash q = q_b$, say
$$q_b \;\cong\; \sum_{i \in I} a_i(r_i) \cdot q_i + \sum_{j \in J} b_j(s_j)$$
Then
$$\partial_H(q_b) \;\cong\; \sum_{i \in I} \partial_H(a_i)(r_i) \cdot \partial_H(q_i) + \sum_{j \in J} \partial_H(b_j)(s_j)$$
Clearly $\partial_H(a_i)(r_i)$ and $\partial_H(q_i)$ and $\partial_H(b_j)(s_j)$ are all equal to process terms that do not contain $\|$, $\mathbin{\|\!\_}$, $|$ and ∂_H. So $\partial_H(q_b)$ has a basic form. □

4.5 Completeness of ACPρ

Theorem 4.4 $\forall p, q \in \mathcal{T} \qquad p \leftrightarrow q \implies$ ACP$\rho \vdash p = q$

Proof. Suppose that $p \leftrightarrow q$. According to Theorem 4.3 there are $p_b, q_b \in \mathcal{B}$ such that $p = p_b$ and $q = q_b$. Then $p_b \leftrightarrow p \leftrightarrow q \leftrightarrow q_b$, and since we have already proven completeness for basic terms we get $p_b = q_b$. Then $p = q$. □

5 Integrals

5.1 Introduction

Integration is the alternative composition over a continuum of alternatives ([BB91]). So if an action a can happen somewhere in the interval $[1,2]$ we write:

$$\int_{v\in[1,2]} a(v)$$

In this section we take a more restrictive view on integration than in [BB91], called *prefixed* integration. We require that every action has as time stamp a time variable directly preceded by the binding integral. Furthermore, integration is only allowed over intervals of which the bounds are linear expressions of variables. E.g. we allow the following terms:

$$\int_{v\in[1,2]} a(v)\cdot\int_{w\in[v+1,v+2]} b(w) \quad\text{and}\quad (\int_{v\in\langle 0,1\rangle} a(v)+\int_{w\in\langle 0,2\rangle} b(w))\cdot\int_{z\in[v,w]} c(z)$$

but not

$$\int_{v\in V}\int_{w\in W} a(w) \quad\text{or}\quad \int_{v\in V} a(2)\cdot b(v) \quad\text{or}\quad \int_{v\in\{1,2\}} a(v)$$

5.2 Bounds and intervals

$TVar$ denotes an infinite, countable set of *time variables.* Let $t\in[0,\infty]$, $r\in\langle 0,\infty\rangle$ and $v\in TVar$. The set *Bound* of *bounds*, with typical element b, is defined by

$$b ::= t \mid v \mid b+b \mid b \dot{-} b \mid r\cdot b$$

where $\dot{-}$ denotes the monus function, i.e. if $t_0\le t_1$ then $t_0\dot{-}t_1=0$. In the sequel $\triangleleft$ and $\triangleright$ are elements of $\{\langle,[\}$ and $\{\rangle,]\}$ respectively. An interval V is of the form $\triangleleft\, b_1,b_2\,\triangleright$ with b_1,b_2 bounds.

For $b\in \mathit{Bound}$ the set of time variables occurring in b is denoted by $tvar(b)$. Of course $tvar(\triangleleft\, b,c\,\triangleright)=tvar(b)\cup tvar(c)$.

For *time-closed* intervals, i.e. intervals for which $tvar(V)=\emptyset$, two operators sup, inf are defined. Let $V=\triangleleft\, t_0,t_1\,\triangleright$

$$\begin{array}{rcl} V\neq\emptyset & : & inf(V)=t_0 \\ & & sup(V)=t_1 \\ V=\emptyset & : & inf(V)=sup(V)=0 \end{array}$$

5.3 Process terms

Let $a\in A_\delta$, $t\in[0,\infty]$, $v\in TVar$, V an interval and b a bound. The set $\mathcal{T}$ of *process terms*, with typical element p, is defined by

$$p ::= a(t) \mid \int_{v\in V} a(v) \mid \int_{v\in V}(a(v)\cdot p) \mid p+p \mid p\cdot p \mid b\gg p$$

In the sequel $\int_{v\in[b,b]} a(v)$ is abbreviated by $a(b)$. Furthermore, we will use a *scope convention*, saying that if we do not write scope brackets, then the scope is as large as possible. Thus we write $\int_{v\in V} a(v)\cdot p$ for $\int_{v\in V}(a(v)\cdot p)$.

5.4 Free time variables

We now define inductively the collection $FV(p)$ of time variables appearing in a process term p that are not bound by an integral sign, the so-called *free* variables:

$$\begin{array}{lcl} FV(\int_{v\in V} a(v)) & = & tvar(V) \\ FV(\int_{v\in V} a(v)\cdot p) & = & (FV(p)\backslash\{v\})\cup tvar(V) \\ FV(p+q) & = & FV(p)\cup FV(q) \\ FV(p\cdot q) & = & FV(p)\cup FV(q) \\ FV(b\gg p) & = & FV(p)\cup tvar(b) \end{array}$$

A term p with $FV(p)=\emptyset$ is called a *time-closed* term.

$$\mathcal{T}^{cl}=\{p\in\mathcal{T}|FV(p)=\emptyset\}$$

5.5 Substitutions

A *substitution* is a mapping from *TVar* to *Bound*. For σ a substitution and b a bound, $\sigma(b)$ denotes the bound that results from substituting $\sigma(v)$ for each occurrence of v in b for all $v\in TVar$. Of course $\sigma(\langle\!\!\langle b_1,b_2\rangle\!\!\rangle)=\langle\!\!\langle\sigma(b_1),\sigma(b_2)\rangle\!\!\rangle$.

Substitutions are extended to process terms by defining five inductive rules. The first four rules are easy:

$$\begin{array}{c} \sigma(p+q)=\sigma(p)+\sigma(q) \\ \sigma(p\cdot q)=\sigma(p)\cdot\sigma(q) \\ \sigma(b\gg p)=\sigma(b)\gg\sigma(p) \\ \sigma(\int_{v\in V} a(v))=\int_{v\in\sigma(V)} a(v) \end{array}$$

The fifth rule, defining $\sigma(\int_{v\in V} a(v)\cdot p)$, is more complicated. First of all, the free occurrences of v in p are bound by the integral sign $\int_{v\in V}$. So these occurrences of v are not to be substituted by $\sigma(v)$. Hence, if σ_v denotes the substitution that is equal to σ on $TVar\backslash\{v\}$ while $\sigma_v(v)=v$, then

$$\sigma(\int_{v\in V} a(v)\cdot p)=\int_{v\in\sigma(V)} a(v)\cdot\sigma_v(p)$$

But there is a second problem; if $w\in FV(p)\backslash\{v\}$, then after substituting $\sigma(w)$ for w in p, all occurrences of v in $\sigma(w)$ are suddenly bound by $\int_{v\in V}$. So this definition of $\sigma(\int_{v\in V} a(v)\cdot p)$ is only valid if we have

$$\forall w\in FV(p)\backslash\{v\}\quad v\notin tvar(\sigma(w))$$

If this requirement does not hold, then the expression $\sigma(\int_{v\in V} a(v)\cdot p)$ is undefined.

If there is only one $v\in TVar$ such that $\sigma(v)\neq v$, then $\sigma(p)$ can be denoted by $p[\sigma(v)/v]$.

5.6 α-conversion

Process terms are considered modulo α-conversion; we extend the definition of $\cong$ as follows.

$$\begin{array}{rcl} \int_{v\in V} a(v) & \cong & \int_{w\in V} a(w) \\ \text{if } w \text{ does not occur in } p\text{, then } \int_{v\in V} a(v)\cdot p & \cong & \int_{w\in V} a(w)\cdot p[w/v] \end{array}$$

We take the transitive closure of this relation.

Note that by applying α-conversion we can always ensure that for $p\in\mathcal{T}$ and σ a substitution, the expression $\sigma(p)$ is well-defined.

5.7 The theory of BPA$\rho\delta I$

In this section we give an axiom system BPA$\rho\delta I$ for time-closed process terms. We have five axioms, called INT0-4, that express the specific properties of terms with integration. The other axioms are the BPA$\rho\delta I$ versions of BPA$\rho\delta$ axioms.

The axiom system for BPA$\rho\delta I$ consists of the contents of Table 7 together with the axioms A1-5 if BPA. In Table 7 it is assumed that $p, q \in \mathcal{T}$ with $FV(p+q) \subseteq \{v\}$ and $X, Y \in \mathcal{T}^{cl}$ and P is of the form $a(v)$ or $a(v) \cdot p$.

Note that the axioms A3 and A5 are not really necessary, because A3 can be deduced from INT1 and A5 from A4 together with INT3.

INT0		$a(t) = \int_{v\in[t,t]} a(v)$
INT1	$V = V_0 \cup V_1$	$\int_{v\in V_0} P + \int_{v\in V_1} P = \int_{v\in V} P$
INT2		$\int_{v\in\emptyset} P = \delta$
INT3a		$\int_{v\in V}(a(v)) \cdot Y = \int_{v\in V} a(v) \cdot Y$
INT3b		$\int_{v\in V}(a(v) \cdot p) \cdot Y = \int_{v\in V} a(v) \cdot (p \cdot Y)$
INT4		$\forall t \in V \;\; X + a(t) \cdot p[t/v] = X \implies X + \int_{v\in V} a(v) \cdot p = X$
ATA1a		$a(0) = \delta$
ATA1b		$a(\infty) = \delta(\infty)$
AI2		$\int_{v\in V} \delta(v) = \delta(sup(V))$
AI3		$\int_{v\in V} \delta(v) \cdot p = \delta(sup(V))$
AI4	$t \leq sup(V)$	$\int_{v\in V} P + \delta(t) = \int_{v\in V} P$
SHI1		$\int_{v\in V} a(v) \cdot p = \int_{v\in V} a(v) \cdot (v \gg p)$
SHI2		$t \gg \int_{v\in V} P = \int_{v\in V\cap(t,\infty]} P + \delta(t)$
SH3		$t \gg (X+Y) = (t \gg X) + (t \gg Y)$

Table 7: An axiom system for BPA$\rho\delta I$

If $p[t/v] = q[t/v]$ for all $t \in V$, then INT4 implies

$$\int_{v\in V} a(v) \cdot p = \int_{v\in V} a(v) \cdot p + \int_{v\in V} a(v) \cdot q = \int_{v\in V} a(v) \cdot q$$

We will refer to this equality as INT4$^-$. Furthermore, BPA$\rho\delta I^-$ will denote the axiom system BPA$\rho\delta I$ with INT4 replaced by INT4$^-$. We will show in Section 9 that BPA$\rho\delta I$ and BPA$\rho\delta I^-$ are equivalent proof systems for the algebra of time-closed process terms.

5.8 Removing the scope brackets

According to the scope convention we can write

$$\int_{v\in<1,2>} (a(v) \cdot \int_{w\in<3,4>} (b(w))) \text{ as } \int_{v\in<1,2>} a(v) \cdot \int_{w\in<3,4>} b(w)$$

and no scope brackets are needed. For other terms however we have to do some work before all *scope* brackets can be removed. Consider the following term, where the time variable v of the last integral is in the scope of the first one.

$$\int_{v\in<0,1>}(a(v)\cdot\int_{v\in<1,2>}(b(v)\cdot\int_{w\in<3,4>}(c(w)))\cdot\int_{x\in<v+5,v+6>}(d(x)))$$

Apply an α-conversion

$$\int_{u\in<0,1>}(a(u)\cdot\int_{v\in<1,2>}(b(v)\cdot\int_{w\in<3,4>}(c(w)))\cdot\int_{x\in<u+5,u+6>}(d(x)))$$

and a *scope widening* according to INT3a,b

$$\int_{u\in<0,1>}(a(u)\cdot\int_{v\in<1,2>}(b(v)\cdot\int_{w\in<3,4>}(c(w)\cdot\int_{x\in<u+5,u+6>}(d(x)))))$$

before removing the brackets according to the *scope convention*

$$\int_{u\in<0,1>}a(u)\cdot\int_{v\in<1,2>}b(v)\cdot\int_{w\in<3,4>}c(w)\cdot\int_{x\in<u+5,u+6>}d(x)$$

We make this more formal by a definition and a proposition.

Definition 5.1 *The set of* widest scope *terms* $\mathcal{W}$ *is defined by*

$$\mathcal{W}=\{p\in\mathcal{T}\mid p \text{ does not contain a subterm of the form } p_0\cdot p_1\}$$

The following proposition can be proven by induction on the size of p, using α-conversion and scope widening.

Proposition 5.2 *For each term* $p\in\mathcal{T}$ *there is a* $p_w\in\mathcal{W}$ *such that* $p=p_w$.

5.9 An operational semantics for BPA$\rho\delta I$

We expect the following transitions:

$$\begin{array}{lcl}\int_{v\in<0,10>}a(v) & \xrightarrow{a(5)} & \surd\\ \int_{v\in<0,10>}a(v)\cdot\int_{w\in<v,v+1>}b(w) & \xrightarrow{a(5)} & \int_{w\in<5,6>}b(w)\end{array}$$

since $5\in<0,10>$.

The atomic rules for BPA$\rho\delta I$ are given in Table 8. We extend the definitions of $U(p)$ and $L(p)$ from BPA$\rho\delta$ to $\mathcal{T}^{cl}$ as follows. Let V be a time-closed interval and let P be of the form $a(v)$ or $a(v)\cdot p$ with $FV(p)\subseteq\{v\}$.

$$U(\int_{v\in V}P)=sup(V)\qquad L(\int_{v\in V}P)=\begin{cases}0 & \text{if } a=\delta\\ sup(V) & \text{otherwise}\end{cases}$$

Bisimulation equivalence is denoted by $\leftrightarrow$ and is defined as in Definition 2.1. In [Klu91] it is proven that bisimulation equivalence is a congruence.

Theorem 5.3 *The theory of* BPA$\rho\delta I$ *is sound.*

The operational semantics presented here characterizes the same bisimulation equivalence as the operational semantics for integration of [BB91].

$a \in A$			
atom :	$t \in V \backslash \{0, \infty\}$	$\int_{v \in V} a(v) \xrightarrow{a(t)} \surd$	
	$t \in V \backslash \{0, \infty\}$	$\int_{v \in V} a(v) \cdot p \xrightarrow{a(t)} t \gg p[t/v]$	

Table 8: Additional action rules for integration

6 Completeness for Integration

6.1 Basic terms

In Definition 3.1 the notion of a basic term for BPA$\rho\delta$ (without integration) has been introduced. In this section we introduce a notion of basic terms for BPA$\rho\delta I$. Remember that the idea of a basic term is that it does not contain "redundant" information. For example:

$$\int_{v \in <0,10>} a(v) \cdot (\int_{w \in <0,10>} b(w) + \int_{z \in <0,10]} \delta(z)) = \int_{v \in <0,10>} a(v) \cdot \int_{w \in <v,10>} b(w)$$

$$a(10) \cdot (\int_{v \in <0,20>} b(v) + \delta(5) + \int_{w \in <0,30>} \delta(w)) = a(10) \cdot (\int_{v \in <10,20>} b(v) + \delta(30))$$

The terms on the left-hand side are not basic terms, since they contain "redundant" information; intervals can be decreased and summands can be removed. The terms on the right-hand side are basic terms.

Definition 6.1 *Let $p, q \in \mathcal{W}$, $a \in A$, V a time-closed interval in $\langle 0, \infty \rangle$, $s \in [0, \infty]$ and $t \in [0, \infty\rangle$.*

$\mathcal{B}$, $\mathcal{B}(t)$ and $\mathcal{B}^s(t)$ are the smallest sets satisfying

$$\begin{array}{rlll}
 & \delta & \in & \mathcal{B}^\infty(t) \\
t < V & \int_{v \in V} a(v) & \in & \mathcal{B}^\infty(t) \\
t < s & \delta(s) & \in & \mathcal{B}^s(t) \\
t < V \ \wedge \ \forall r \in V (p[r/v] \in \mathcal{B}(r)) & \int_{v \in V} a(v) \cdot p & \in & \mathcal{B}^\infty(t) \\
p \in \mathcal{B}^s(t) \ \wedge \ q \in \mathcal{B}^s(t) & p + q & \in & \mathcal{B}^s(t) \\
p \in \mathcal{B}^s(t) \ \wedge \ q \in \mathcal{B}^\infty(t) \ \wedge \ s > U(q) & p + q, q + p & \in & \mathcal{B}^s(t) \\
 & & & \\
 & \mathcal{B}(t) & = & \cup_s \mathcal{B}^s(t) \\
 & \mathcal{B} & = & \mathcal{B}(0)
\end{array}$$

Note that $\mathcal{B} \subseteq \mathcal{T}^{cl}$ and that basic terms have ascending time stamps.

6.2 Dealing with free time variables

Until now we have mostly considered time-closed terms. But if we want to prove that every time-closed term has a basic form, we have to consider terms with free time variables as well.

Therefore we will introduce the notion of a *conditional* term. A conditional term determines for each substitution of real values for the free time variables a time-closed term. For example, if we consider the term $a(5) \cdot b(v)$, which has a free time variable v, we will associate to it the following conditional term:

if the context assigns a value $t \leq 5$ to v, then deliver $a(5) \cdot \delta$
if the context assigns a value $t > 5$ to v, then deliver $a(5) \cdot b(t)$

This conditional term will be denoted as follows (the notation $:\rightarrow$ is taken from [BB90]):

$$p \cong \{v \leq 5 :\rightarrow a(5) \cdot \delta\} + \{v > 5 :\rightarrow a(5) \cdot b(v)\}$$

For every substitution σ which assigns a real number to v we have $\sigma(p) \in \mathcal{B}$. In the following we introduce a generalization of a basic term, the so-called *basic* conditional term. The idea is that a conditional term is basic if any substitution of reals for the free time variables of the conditional term yields a basic term. These new notions will now be defined more formally.

6.3 Conditions

The set $Cond^{at}$ of *atomic conditions* is defined by

$$Cond^{at} \;:=\; \{b_0 < b_1,\ b_0 \leq b_1,\ b_0 > b_1,\ b_0 \geq b_1,\ b_0 = b_1 \mid b_0, b_1 \in Bound\} \cup \{tt, ff\}$$

where tt denotes 'true' and ff 'false'.

A *condition* consists of conjunctions, disjunctions and negations of atomic conditions. Let $\alpha^{at} \in Cond^{at}$. Then the set of conditions $Cond$, with typical element α, is defined inductively by

$$\alpha \;::=\; \alpha^{at} \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \neg\alpha$$

We denote the set of time variables of $\alpha \in Cond$ by $tvar(\alpha)$.

Let Σ denote the collection of substitutions $\sigma : TVar \rightarrow [0, \infty]$. For $\alpha \in Cond$ we define a subset $[\alpha]$ of Σ.

$$[\alpha] \;:=\; \{\sigma \in \Sigma \mid \sigma(\alpha) \text{ is true}\}$$

Clearly $[tt] = \Sigma$ and $[ff] = \emptyset$.

A finite collection of conditions $\{\alpha_1, ..., \alpha_n\}$ is called a *partition* if for each $\sigma \in \Sigma$ there is exactly one i such that $\sigma \in [\alpha_i]$. A collection of conditions $\{\beta_j\}$ is called a *refinement* of a collection of conditions $\{\alpha_i\}$ if $\cup_j[\beta_j] = \cup_i[\alpha_i]$ and for each j there is an i such that $[\beta_j] \subseteq [\alpha_i]$.

In the sequel we will often abbreviate conditions by expressions like $b \in V$, $V \cap W \neq \emptyset$, $sup(V) > b$, $inf(V) < b$, $U(p) > b$ etc. For example:

- $b \in \langle b_0, b_1]$ abbreviates $b_0 < b \wedge b \leq b_1$
- $\langle b_0, b_1] \cap [c_0, c_1] = \emptyset$ abbreviates $b_0 \geq b_1 \vee c_0 > c_1 \vee b_1 < c_0 \vee b_0 \geq c_1$
- $sup(\langle b_0, b_1]) < b$ abbreviates $b_1 < b \;\vee\; (b_0 \geq b_1 \wedge b > 0)$

6.4 Conditional terms

Let $\alpha \in Cond$, $p \in \mathcal{T}$, $a \in A_\delta$ and $b \in Bound$. The set $\mathcal{C}$ of *conditional* terms, with typical element p_c, is defined by

$$p_c \ ::= \ p \ | \ \alpha :\rightarrow p_c \ | \int_{v\in V} a(v)\cdot p_c \ | \ p_c + p_c \ | \ b \gg p_c$$

On $\mathcal{C}$ the equivalence $\cong$ means syntactic equivalence modulo α-conversion and modulo associativity and commutativity of the $+$.

Equalities on $\mathcal{C}$ may be hard to read, e.g. $b_0 = b_1 :\rightarrow p = c_0 = c_1 :\rightarrow q$. Therefore we will write conditional terms between brackets: $\{b_0 = b_1 :\rightarrow p\} = \{c_0 = c_1 :\rightarrow q\}$. These brackets are not part of the syntax but are meta-brackets.

Often conditional terms will be abbreviated. For instance, let V be the interval $[b_0, b_1)$ and b a bound. Then

$$\{b \geq sup(V) :\rightarrow \delta\} + \{b < sup(V) :\rightarrow \int_{v\in V\cap(b,\infty]} P\}$$

abbreviates

$$\{b_0 \geq b_1 \vee b \geq b_1 :\rightarrow \delta\} \ + \ \{b_0 \leq b < b_1 :\rightarrow \int_{v\in(b,b_1)} P\} \ + \ \{b < b_0 < b_1 :\rightarrow \int_{v\in[b_0,b_1)} P\}$$

6.5 The theory of CTA

The equational theory CTA (Conditional Terms Algebra), given in Table 9 consists of 'conditional variants' of axioms of BPA$\rho\delta I$ together with six new axioms CON1-6. Using these conditional axioms it is possible to reason about terms containing free time variables. We will use the axioms of CTA to reduce each process term to a basic conditional term.

In order to use CON6 for general conditional terms we need the following lemma. Note that it would not be true if we would allow expressions like v^2 to be bounds.

Lemma 6.2 *Let $v \in TVar$. A condition α always has a refinement of the form $\{\beta_j \wedge v \in V_j\}$, where $tvar(\beta_j) \cup tvar(V_j) \subseteq tvar(\alpha)\backslash\{v\}$.*

Proof. It is easy to see that α can be rewritten to a condition $\vee_i \gamma_i$ with each γ_i of the form

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

where v does not occur in the γ, b_j, c_k, d_l.

We show that each γ_i is equivalent to a condition $\vee_j(\beta_j \wedge v \in V_j)$ with $v \notin tvar(\beta_j) \cup tvar(V_j)$. Then clearly we are done. Fix an i.

- If $J \cup K \cup L = \emptyset$, then $\gamma \wedge v \in [0,\infty]$ is equal to γ_i.
- Let $L \neq \emptyset$. Fix an $l_0 \in L$ and put $d = d_{l_0}$. Then the condition

$$(\gamma \wedge \bigwedge_{j\in J} d > b_j \wedge \bigwedge_{k\in K} d < c_k \wedge \bigwedge_{l\in L} d = d_l) \ \wedge \ v \in [d,d]$$

  is equal to γ_i. So we can assume $L = \emptyset$.

CI1	$\int_{v\in V} P = \{V \subseteq \{0\} :\rightarrow \delta\} + \{V = \{\infty\} :\rightarrow \delta(\infty)\} + \{V \not\subseteq \{0,\infty\} :\rightarrow \int_{v\in V\setminus\{0,\infty\}} P\}$
CI2a	$\int_{v\in V} \delta(v) = \delta(sup(V))$
CI2b	$\int_{v\in V} \delta(v) \cdot p = \delta(sup(V))$
CI3	$\int_{v\in V} P + \delta(b) = \{b \leq sup(V) :\rightarrow \int_{v\in V} P\} + \{b > sup(V) :\rightarrow \int_{v\in V} P + \delta(b)\}$
ATI4	$\int_{v\in V} a(v) \cdot p = \int_{v\in V} a(v) \cdot (v \gg p)$
CBI1	$b \gg \int_{v\in V} P = \{b < sup(V) :\rightarrow \int_{v\in V\cap(b,\infty]} P\} + \{b \geq sup(V) :\rightarrow \delta(b)\}$
ATB2	$b \gg (p_c + q_c) = b \gg p_c + b \gg q_c$
CON1	$\{tt :\rightarrow p\} = p$
CON2	$\{ff :\rightarrow p\} = \delta$
CON3	$\{\alpha :\rightarrow \Sigma_i\{\beta_i :\rightarrow p_i\}\} = \Sigma_i\{\alpha \wedge \beta_i :\rightarrow p_i\}$
CON4	$\{\alpha :\rightarrow p\} + \{\beta :\rightarrow q\} = \{\alpha \wedge \beta :\rightarrow p + q\} + \{\alpha \wedge \neg\beta :\rightarrow p\} + \{\neg\alpha \wedge \beta :\rightarrow q\}$
CON5	$b \gg \{\alpha :\rightarrow p\} = \{\alpha :\rightarrow b \gg p\} + \{\neg\alpha :\rightarrow \delta(b)\}$
CON6	If $\{\alpha_i \wedge v \in W_i\}$ is a partition and $v \notin tvar(\alpha_i) \cup tvar(W_i)$, then $\int_{v\in V} a(v) \cdot \Sigma_i\{\alpha_i \wedge v \in W_i :\rightarrow p_i\} = \Sigma_i\{\alpha_i :\rightarrow \int_{v\in V\cap W_i} a(v) \cdot p_i\}$

Table 9: An axiom system for CTA

- If $J \neq \emptyset$ and $K = \emptyset$, then we can take

$$\bigvee_{j\in J}(\gamma \wedge \bigwedge_{j'\in J} b_j \geq b_{j'} \wedge v \in \langle b_j, \infty])$$

Similarly, if $J = \emptyset$ and $K \neq \emptyset$, then we can take

$$\bigvee_{k\in K}(\gamma \wedge \bigwedge_{k'\in K} c_k \leq c_{k'} \wedge v \in [0, c_k\rangle)$$

And if $J \neq \emptyset$ and $K \neq \emptyset$, then we can take

$$\bigvee_{(j,k)\in J\times K}(\gamma \wedge \bigwedge_{j'\in J} b_j \geq b_{j'} \wedge \bigwedge_{k'\in K} c_k \leq c_{k'} \wedge v \in \langle b_j, c_k\rangle)$$

Thus α is equivalent to a condition $\vee_j(\beta_j \wedge v \in V_j)$ with $v \notin tvar(\beta_j) \cup tvar(V_j)$. □

6.6 Relating CTA and BPA$\rho\delta I^-$

Proposition 6.3 $\forall p, q \in \mathcal{T}^{cl}$ CTA $\vdash p = q \implies$ BPA$\rho\delta I^- \vdash p = q$

Proof. Applying axioms CON1-6, one can easily define an algorithm that reduces each conditional term p_c to the form $\{\alpha_i :\rightarrow p_i\}$, where $\{\alpha_i\}$ is a partition. For each $\sigma \in \Sigma$, let $p_c{<}\sigma{>}$ denote the time-closed process term $\sigma(p_{i(\sigma)})$, where $i(\sigma)$ is such that $\sigma \in [\alpha_{i(\sigma)}]$. We now prove

$$\text{CTA} \vdash p_c = q_c \implies \forall \sigma \in \Sigma \quad \text{BPA}\rho\delta I^- \vdash p_c{<}\sigma{>} = q_c{<}\sigma{>}$$

Then we are done, since for all $p \in \mathcal{T}^{cl}$ and $\sigma \in \Sigma$ we have $p{<}\sigma{>} \equiv p$.

A deduction in CTA consists of equalities of the form $C[p_c] = C[q_c]$, where $C[]$ is a *context* (i.e. a conditional term containing exactly one occurrence of the empty symbol $[]$) and $p_c = q_c$ is an instantiation of an axiom of CTA (i.e. the result of substituting conditional terms for the variables in an axiom). For such an equation $C[p_c] = C[q_c]$ we prove that

$$\forall \sigma \in \Sigma \quad \text{BPA}\rho\delta I^- \vdash C[p_c]{<}\sigma{>} = C[q_c]{<}\sigma{>}$$

using induction to the size of $C[]$. Then we are done.

First assume that $C[]$ has size 1, i.e. $C[] \equiv []$. Then we need to show that for each instantiation $p_c = q_c$ of an axiom of CTA and for each $\sigma \in \Sigma$ we have BPA$\rho\delta I^- \vdash p_c{<}\sigma{>} = q_c{<}\sigma{>}$. It is left to the reader to check that this is indeed the case.

Now suppose that the case has been proven if the size of the context is $\leq n$ and let $C[]$ have size $n + 1$. We distinguish three cases.

1. $C[]$ is of the form $\{\alpha :\rightarrow C'[]\}$.

Fix a $\sigma \in \Sigma$. For each conditional term $\tilde{p}_c$ we have

$$C[\tilde{p}_c]{<}\sigma{>} \equiv \begin{cases} \delta & \text{if } \sigma \notin [\alpha] \\ C'[\tilde{p}_c]{<}\sigma{>} & \text{if } \sigma \in [\alpha] \end{cases}$$

So if $\sigma \notin [\alpha]$ then $C[p_c]{<}\sigma{>} \equiv \delta \equiv C''[q_c]{<}\sigma{>}$, and if $\sigma \in [\alpha]$ then by induction

$$C[p_c]{<}\sigma{>} \equiv C'[p_c]{<}\sigma{>} = C'[q_c]{<}\sigma{>} \equiv C[q_c]{<}\sigma{>}$$

2. $C[]$ is of the form $C'[] + \tilde{p}_c$.
Then $C[p_c]{<}\sigma{>} \equiv C'[p_c]{<}\sigma{>} + \tilde{p}_c{<}\sigma{>}$ and $C[q_c]{<}\sigma{>} \equiv C'[q_c]{<}\sigma{>} + \tilde{p}_c{<}\sigma{>}$. Now the case follows by induction.
3. $C[]$ is of the form $\int_{v\in V} a(v)\cdot C'[]$.
Let $\sum_i\{\alpha_i :\to p_i\}$ and $\sum_j\{\beta_j :\to q_j\}$ be the results of applying the algorithm mentioned at the beginning of this proof to $C'[p_c]$ and $C'[q_c]$ respectively. Using the construction from Lemma 6.2, $\{\alpha_i\}$ and $\{\beta_j\}$ can be reduced to partitions $\{\alpha'_k \wedge v \in V_k\}_{k\in K}$ resp. $\{\beta'_l \wedge v \in W_l\}_{l\in L}$, where v does not occur in the α'_k, β'_l, V_k, W_l. Let $[\alpha'_k \wedge v \in V_k] \subseteq [\alpha_{i_k}]$ and $[\beta'_l \wedge v \in W_l] \subseteq [\beta_{j_l}]$. Fix a $\sigma \in \Sigma$. As before σ_v denotes σ restricted to $TVar\backslash\{v\}$. Furthermore, σ_t denotes the substitution that is equal to σ on $TVar\backslash\{v\}$ while $\sigma_t(v) = t$. Let K_σ resp. L_σ denote the collection of $k \in K$ resp. $l \in L$ for which $\sigma(\alpha'_k)$ resp. $\sigma(\beta'_l)$ is true. Then

$$C[p_c]{<}\sigma{>} \equiv \sum_{k\in K_\sigma} \int_{v\in\sigma(V)\cap\sigma(V_k)} a(v)\cdot\sigma_v(p_{i_k})$$

$$C[q_c]{<}\sigma{>} \equiv \sum_{l\in L_\sigma} \int_{v\in\sigma(V)\cap\sigma(W_l)} a(v)\cdot\sigma_v(q_{j_l})$$

Since $\{\alpha'_k \wedge v \in V_k\}_{k\in K}$ and $\{\beta'_l \wedge v \in W_l\}_{l\in L}$ are partitions, it follows that for each $t \in [0,\infty]$ there are unique $k(t) \in K_\sigma$ and $l(t) \in L_\sigma$ such that $t \in \sigma(V_{k(t)}) \cap \sigma(W_{l(t)})$. Then clearly $C'[p_c]{<}\sigma_t{>} \equiv \sigma_t(p_{i_{k(t)}})$ and $C'[q_c]{<}\sigma_t{>} \equiv \sigma_t(q_{j_{l(t)}})$. Now the induction hypothesis implies

$$\sigma_v(p_{i_{k(t)}})[t/v] \equiv \sigma_t(p_{i_{k(t)}}) = \sigma_t(q_{j_{l(t)}}) \equiv \sigma_v(q_{j_{l(t)}})[t/v]$$

This holds for all t, so INT4$^-$ induces $C[p_c]{<}\sigma{>} = C[q_c]{<}\sigma{>}$. □

Without the restraint in axiom CON6 that $\{\alpha_i \wedge v \in W_i\}$ is a partition, the previous proposition would not hold. For then we would get equalities like

$$\text{CTA} \vdash \int_{v\in V} a(v)\cdot\delta = \delta$$
$$\text{CTA} \vdash \int_{v\in V} a(v)\cdot(p+q) = \int_{v\in V} a(v)\cdot p + \int_{v\in V} a(v)\cdot q$$

6.7 Basic conditional terms

The collection $\mathcal{B}_c$ of *basic* conditional terms is defined as follows. A conditional term $\Sigma_i\{\alpha_i :\to p_i\}$ is in $\mathcal{B}_c$ if

1. $\{\alpha_i\}$ is a partition
2. $\forall\sigma \in [\alpha_i] \quad \sigma(p_i) \in \mathcal{B}$

For example, $\int_{v\in<b,10>} a(v)$ can be rewritten to the basic conditional term

$$\{b < 10 :\to \int_{v\in<b,10>} a(v)\} + \{b \geq 10 :\to \delta\}$$

The next lemma states that every term in $\mathcal{W}$ (with possibly free time variables) can be rewritten to a basic conditional term.

Lemma 6.4 *For each $p \in \mathcal{W}$ there is a $p_c \in \mathcal{B}_c$ such that*

$$FV(p_c) \subseteq FV(p) \quad \wedge \quad \mathrm{CTA} \vdash p = p_c$$

Proof. By induction on the *size* of p. Let $a \in A$. It is sufficient to consider the following four cases.

1.

$$\int_{v \in V} a(v) = \begin{array}{lll} \{V \not\subseteq \{0,\infty\} & :\rightarrow & \int_{v \in V \setminus \{0,\infty\}} a(v)\} \\ + & & \\ \{V \subseteq \{0\} & :\rightarrow & \delta\} \\ + & & \\ \{V = \{\infty\} & :\rightarrow & \delta(\infty)\} \end{array}$$

2.

$$\begin{array}{ll} & p + q \\ = & \sum_i \{ \alpha_i :\rightarrow p_i \} + \sum_j \{ \beta_j :\rightarrow q_j \} \quad \text{by induction} \\ = & \sum_{(i,j)} \{ \alpha_i \wedge \beta_j :\rightarrow p_i + q_j \} \quad \text{since } \{\alpha_i\} \text{ and } \{\beta_j\} \text{ are partitions} \\ = & \sum_{(i,j)} \left\{ \begin{array}{lll} \alpha_i \wedge \beta_j \wedge U(p_i + q_j) > U(\overline{p_i} + \overline{q_j}) & :\rightarrow & \overline{p_i} + \overline{q_j} + \delta(U(p_i + q_j)) \\ + & & \\ \alpha_i \wedge \beta_j \wedge U(p_i + q_j) \leq U(\overline{p_i} + \overline{q_j}) & :\rightarrow & \overline{p_i} + \overline{q_j} \end{array} \right\} \end{array}$$

where $\overline{p}$ is the term p without δ-summands.

3.

$$\begin{array}{lll} b \gg p & = & b \gg \sum_i \{\alpha_i :\rightarrow \sum_j \int_{v \in V_j} P_j\} \quad \text{by induction} \\ & = & \sum_i \{\alpha_i :\rightarrow b \gg \sum_j \int_{v \in V_j} P_j\} \\ & = & \sum_{(i,j)} \left\{ \begin{array}{lll} \alpha_i \wedge b < sup(V_j) & :\rightarrow & \int_{v \in V_j \cap \langle b, \infty\rangle} P_j \\ + & & \\ \alpha_i \wedge b \geq sup(V_j) & :\rightarrow & \delta(b) \end{array} \right\} \end{array}$$

Since we have to end up with a basic conditional term we have to construct a partition. This can be done by applying sufficiently many times axioms CON4 and CBI1.

4.

$$\begin{array}{ll} & \int_{v \in V} a(v) \cdot p \\ = & \int_{v \in V} a(v) \cdot (v \gg p) \\ = & \int_{v \in V} a(v) \cdot \sum_i \{\alpha_i \wedge v \in W_i :\rightarrow p_i\} \quad \text{by the previous case and Lemma 6.2} \\ = & \sum_i \{\alpha_i :\rightarrow \int_{v \in V \cap W_i} a(v) \cdot p_i\} \end{array}$$

$$\begin{array}{ll} & \Sigma_i\{\alpha_i \wedge V \cap W_i \not\subseteq \{0,\infty\} :\rightarrow \int_{v\in V\cap W_i\setminus\{0,\infty\}} a(v)\cdot p_i\} \\ & + \\ = & \Sigma_i\{\alpha_i \wedge V \cap W_i \subseteq \{0\} :\rightarrow \delta\} \\ & + \\ & \Sigma_i\{\alpha_i \wedge V \cap W_i = \{\infty\} :\rightarrow \delta(\infty)\} \end{array}$$

Some of these p_i's may be of the form $\delta(v)$ and must be rewritten to δ. The conditions are reduced to a partition by applying axioms CON4 and CBI1. □

Theorem 6.5 *For each $p \in \mathcal{T}^{cl}$ there is a $p_b \in \mathcal{B}$ such that* BPA$\rho\delta I^-\vdash p = p_b$.

Proof. There is a $p' \in \mathcal{W}$ such that BPA$\rho\delta I^-\vdash p = p'$. By the previous lemma there is a $p_c \in \mathcal{B}_c$ with $FV(p_c) \subseteq FV(p') = \emptyset$ and CTA$\vdash p' = p_c$. Since p_c does not contain free time variables, all conditions occurring in p_c are either true or false. Then clearly there is a time-closed process term $p_b \in \mathcal{B}$ such that CTA$\vdash p_c = p_b$. So according to Proposition 6.3 BPA$\rho\delta I^-\vdash p = p_b$. □

6.8 Completeness of BPA$\rho\delta I$

Theorem 6.6 $\forall p, q \in \mathcal{T}^{cl} \qquad p \leftrightarrow q \implies$ BPA$\rho\delta I \vdash p = q$

Proof. Theorem 6.5 implies that it is sufficient to consider basic terms only. We will prove by induction on the depth of p that $p \subseteq q$. Assume

$$p \cong \sum_i \int_{v\in V_i} a_i(v)\cdot p_i + \sum_j \int_{v\in W_j} b_j(v)$$

Since p is a basic term there is for each j and each $t \in W_j$ a transition $p \xrightarrow{b_j(t)} \surd$. We assume $p \leftrightarrow q$, so there is for each j and each $t \in W_j$ a transition $q \xrightarrow{b_j(t)} \surd$ and thus $b_j(t) \subseteq q$. It follows that

$$\int_{v\in W_j} b_j(v) \subseteq q$$

Similarly there is for each i and each $t \in V_i$ a term q' such that $q \xrightarrow{a_i(t)} t \gg q'$ and $t \gg p_i[t/v] \leftrightarrow t \gg q'$. Since p and q are basic terms we have

$$p_i[t/v] \leftrightarrow t \gg p_i[t/v] \leftrightarrow t \gg q' \leftrightarrow q'$$

and thus by induction it follows that $p_i[t/v] = q'$. Together with $a_i(t)\cdot q' \subseteq q$ we may conclude

$$a_i(t)\cdot p_i[t/v] \subseteq q$$

Now we can apply axiom INT4 to get $\int_{v\in V_i} a_i(v)\cdot p_i \subseteq q$. □

M1		$X\|Y = X⌊\!⌊Y + Y⌊\!⌊X + X\|Y$
LMI1a	$inf(V) < U(Y)$	$\int_{v\in V} a(v)⌊\!⌊Y = \int_{v\in V\cap\langle 0,U(Y)\rangle} a(v)\cdot Y$
LMI1b	$inf(V) \geq U(Y)$	$\int_{v\in V} a(v)⌊\!⌊Y = \delta(U(Y))$
LMI2a	$inf(V) < U(Y)$	$\int_{v\in V}(a(v)\cdot p)⌊\!⌊Y = \int_{v\in V\cap\langle 0,U(Y)\rangle} a(v)\cdot(p\|Y)$
LMI2b	$inf(V) \geq U(Y)$	$\int_{v\in V}(a(v)\cdot p)⌊\!⌊Y = \delta(U(Y))$
LM3		$(X+Y)⌊\!⌊Z = X⌊\!⌊Z + Y⌊\!⌊Z$
CMI1a	$V_0 \cap V_1 \neq \emptyset$	$\int_{v\in V_0} a(v)\| \int_{v\in V_1} b(v) = \int_{v\in V_0\cap V_1}(a\|b)(v)$
CMI1b	$V_0 \cap V_1 = \emptyset$	$\int_{v\in V_0} P\| \int_{v\in V_1} P' = \delta(min\{sup(V_0), sup(V_1)\})$
CMI2	$V_0 \cap V_1 \neq \emptyset$	$\int_{v\in V_0}(a(v)\cdot p)\| \int_{v\in V_1} b(v) = \int_{v\in V_0\cap V_1}(a\|b)(v)\cdot p$
CMI3	$V_0 \cap V_1 \neq \emptyset$	$\int_{v\in V_0} a(v)\| \int_{v\in V_1}(b(v)\cdot q) = \int_{v\in V_0\cap V_1}(a\|b)(v)\cdot q$
CMI4	$V_0 \cap V_1 \neq \emptyset$	$\int_{v\in V_0}(a(v)\cdot p)\| \int_{v\in V_1}(b(v)\cdot q) = \int_{v\in V_0\cap V_1}(a\|b)(v)\cdot(p\|q)$
CM5		$(X+Y)\|Z = X\|Z + Y\|Z$
CM6		$X\|(Y+Z) = X\|Y + X\|Z$
DI1		$\partial_H(\int_{v\in V} a(v)) = \int_{v\in V}\partial_H(a)(v)$
DI2		$\partial_H(\int_{v\in V} a(v)\cdot p) = \int_{v\in V}\partial_H(a)(v)\cdot\partial_H(p)$
D3		$\partial_H(X+Y) = \partial_H(X) + \partial_H(Y)$

Table 10: An axiom system for ACPρI

7 Adding Communication

In this section we incorporate the operators $\|, |, ⌊\!⌊$ and ∂_H into the theory BPA$\rho\delta I$. First we have to adapt the definition of our set of terms $\mathcal{T}$ by adding the rules

$$\begin{array}{lll} p,q\in\mathcal{T} & \Longrightarrow & p\Box q\in\mathcal{T} \quad \Box\in\{\|,|,⌊\!⌊\} \\ p\in\mathcal{T} & \Longrightarrow & \partial_H(p)\in\mathcal{T} \end{array}$$

The theory of ACPρI consists of adapted axioms of ACPρ as given in Table 5.

We can gather the atom rules from Table 8, the rules for $\cdot, +, \gg$ and δ from Table 4 and the rules for $\|, |, ⌊\!⌊$ and ∂_H from Table 6 to define a an operational semantics for ACPρI. The definition of $L(p)$ from ACPρ is extended to ACPρI as follows:

$$initact(\int_{v\in V} P) = \begin{cases} \emptyset & \text{if } a=\delta \\ \{a(t) \mid t\in V\backslash\{0,\infty\}\} & \text{otherwise} \end{cases}$$

Using Definition 2.1 we define bisimulation equivalence for ACPρI. This equivalence is a congruence ([Klu91]).

Theorem 7.1 *The theory of* ACPρI *is sound.*

The operational semantics for ACPρI that we defined here is equivalent to the original one from [BB91].

The following theorem can be proven by extending the notion of a conditional term to ACPρ and defining extra conditional axioms. It says that the merge, left merge and communication merge can be eliminated.

Theorem 7.2 *For each $p \in \mathcal{T}^{cl}$ there is a $p_b \in \mathcal{B}$ such that* BPA$\rho\delta I \vdash p = p_b$

Using this theorem we can prove that ACPρI is complete w.r.t. bisimulation equivalence.

8 Reducing Process Terms

In Section 5 we have proven that each time-closed process term is equal to a basic term. However, this basic form is by no means unique. For instance, if p is a basic term, then $p+p$ is one too.

In this section the machinery is introduced to reduce each process term p to a normal form $p\!\downarrow$. Process terms are considered modulo commutativity and associativity of the $+$. For each time-closed process term its normal form will again be a time-closed process term. And in the next section we will prove that if $p, q \in \mathcal{T}^{cl}$ with $p = q$, then $p\!\downarrow \cong q\!\downarrow$. Thus it is possible to check in a finite number of steps if two terms $p, q \in \mathcal{T}^{cl}$ are equal or not; first p and q are reduced to normal form, and then a finite computation is carried out to see if A1,2 $\vdash p\!\downarrow = q\!\downarrow$ or not.

In the following paragraphs a number of *rewrite rules* are defined, which will be expressions of the form $p_c \longrightarrow q_c$ with $p_c, q_c \in \mathcal{C}$. Paragraph 8.5 will contain an explicit description how to reduce a basic conditional term to a normal form using the rewrite rules. This normal form will again be a basic conditional term.

8.1 Reducing conditions

We define four rewrite rules that reduce conditions. The first three are

1. $\{\alpha :\rightarrow \Sigma_i\{\beta_i :\rightarrow p_i\}\} \longrightarrow \Sigma_i\{\alpha \wedge \beta_i :\rightarrow p_i\}$
2. $\{\alpha :\rightarrow p\} + \{\beta :\rightarrow q\} \longrightarrow \{\alpha \wedge \beta :\rightarrow p + q\} + \{\alpha \wedge \neg\beta :\rightarrow p\} + \{\neg\alpha \wedge \beta :\rightarrow q\}$
3. $\{\alpha :\rightarrow p\} + q \longrightarrow \{\alpha :\rightarrow p + q\} + \{\neg\alpha :\rightarrow q\}$

The fourth rule is based on axiom CON6. Let $\{\alpha_i\}$ be a partition and $v \in TVar$. Using the construction from the proof of Lemma 6.2, $\{\alpha_i\}$ can be reduced to a partition $\{\beta_j \wedge v \in W_j\}$, where $[\beta_j \wedge v \in W_j] \subseteq [\alpha_{i(j)}]$ for some $i(j)$. Furthermore, $tvar(\beta_j) \cup tvar(W_j) \subseteq tvar(\alpha_{i(j)})$. We define

4. $\int_{v \in V} a(v) \cdot \Sigma_i\{\alpha_i :\rightarrow p_i\} \longrightarrow \Sigma_j\{\beta_j :\rightarrow \int_{v \in V \cap W_j} a(v) \cdot p_{i(j)}\}$

8.2 Reducing bounds

In order to reduce time-closed process terms to a unique normal form, it is necessary to reduce bounds. For example, the equation $2v \dot{-} 1 = (v \dot{-} 1) + v$ holds for $v \geq 1$, but is untrue for $0 \leq v < 1$. So the equality

$$\int_{v\in\langle t,\infty\rangle} a(v)\cdot\int_{w\in\langle 2v\dot{-}1,2v\rangle} b(w) \;=\; \int_{v\in\langle t,\infty\rangle} a(v)\cdot\int_{w\in\langle (v\dot{-}1)+v,2v\rangle} b(w)$$

holds for $t \geq 1$, but not for $0 \leq t < 1$.

We want to reduce each bound to a normal form. Therefore it is necessary that each element of our time domain has a unique finite representation. This is clearly not the case for the collection of real numbers. As a time domain we assume from now on a countable subset D of $[0,\infty]$ such that

- $\mathbb{Q}_{\geq 0} \cup \{\infty\} \subseteq D$
- $+, \dot{-} : D \times D \to D$
- $\cdot : \mathbb{Q}_{>0} \times D \to D$

The definition of a bound becomes

$$b ::= t \mid v \mid b+b \mid b \dot{-} b \mid r\cdot b$$

where $t \in D$, $r \in \mathbb{Q}_{>0}$ and $v \in TVar$.

In Appendix A the notion of a bound is generalized to that of a *conditional* bound, which allows expressions of the form $\{\alpha :\rightarrow b\}$ with α a condition and b a bound. Furthermore, it is described how a bound can be reduced to a normal form, which is a conditional bound with all its monus signs replaced by minus signs. This is done by applying the rewrite rule

$$b \dot{-} c \;\longrightarrow\; \{b > c :\rightarrow b - c\} \;+\; \{b \leq c :\rightarrow 0\}$$

Now we can give rewrite rule 1, which enables us to reduce the bounds occurring in a process term. Let b be a bound that, using the construction described in Appendix A, is reduced to $\Sigma_i\{\alpha_i :\rightarrow b_i\}$. Then

5. $\int_{v\in\langle b,c\rangle} P \;\longrightarrow\; \Sigma_i\{\alpha_i :\rightarrow \int_{v\in\langle b_i,c\rangle} P\}$

We have a similar rewrite rule for if b has normal form $b\!\downarrow$ and symmetric rules for if c has normal form $\Sigma_i\{\alpha_i :\rightarrow c_i\}$ or $c\!\downarrow$.

8.3 Substituting redundant variables

A variable occurring in a process term can be *redundant* in the sense that only one value can be substituted for it. For example:

$$\int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) = \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w)$$

So in order to reduce time-closed process terms to a unique normal form, it is necessary to substitute the only possible value for such a redundant variable.

The following rewrite rule reduces process terms of the form $\int_{v\in[b,b]} a(v)\cdot p$. Ensure by applying α-conversion that $v \notin tvar(b)$ and also that none of the variables in $tvar(b)$ are bound by integral signs occurring in p. Then

$$6.\quad \int_{v\in[b,b]} a(v)\cdot p \longrightarrow \int_{v\in[b,b]} a(v)\cdot p[b/v]$$

8.4 Reducing double terms

The main problem of reducing time-closed process terms to a unique normal form is getting rid of the 'double terms'. We first give two rewrite rules, based on axiom INT1, to deal with this problem.

Let '$V_0 \sim V_1$' denoting that '$V_0 \cup V_1$ is an interval'. Note that this is a condition, i.e. it can be described by a finite number of (in)equalities between bounds.

$$7a.\quad \int_{v\in V_0} a(v) + \int_{v\in V_1} a(v) \longrightarrow \{V_0 \sim V_1 :\rightarrow \int_{v\in V_0\cup V_1} a(v)\} + \{V_0 \not\sim V_1 :\rightarrow \int_{v\in V_0} a(v) + \int_{v\in V_1} a(v)\}$$

$$7b.\quad \int_{v\in V_0} a(v)\cdot p + \int_{v\in V_1} a(v)\cdot p \longrightarrow \{V_0 \sim V_1 :\rightarrow \int_{v\in V_0\cup V_1} a(v)\cdot p\} + \{V_0 \not\sim V_1 :\rightarrow \int_{v\in V_0} a(v)\cdot p + \int_{v\in V_1} a(v)\cdot p\}$$

However, this rule is not sufficient in all cases. Consider the following two examples.

Example 8.1

$$\int_{v\in\langle 0,1\rangle} a(v)\cdot \int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,2\rangle} a(v)\cdot \int_{w\in\langle v,2\rangle} b(w)$$
$$= \int_{v\in\langle 0,1]} a(v)\cdot \int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in\langle 1,2\rangle} a(v)\cdot \int_{w\in\langle v,2\rangle} b(w)$$

Although these terms are equal, they can not be rewritten by rule 7*b*.

Example 8.2

$$\int_{v\in\langle 0,2\rangle} a(v)\cdot \int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,1]} a(v)\cdot \int_{w\in\langle 1,2\rangle} b(w) = \int_{v\in\langle 0,2\rangle} a(v)\cdot \int_{w\in\langle v,v+1\rangle} b(w)$$

Again both terms can not be rewritten by rule 7*b*.

A logical solution for avoiding such situations seems to be allowing only integration over open intervals and over intervals consisting of one point. However, the following example shows that this restraint does not work.

Example 8.3

$$\int_{v\in\langle 0,1\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w) + \int_{v\in\langle 1,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

$$= \int_{v\in\langle 0,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

Both terms are equal and satisfy the restraint on intervals, but they can not be rewritten by rule 7*b*. We therefore introduce two rewrite rules to deal with Examples 8.1 and 8.2.

Example 8.1 shows that we need a reduction

$$\int_{v\in\langle b,c\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q \longrightarrow \int_{v\in[b,c\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q$$

if the following two statements are true:

1. $b\in V$.

2. $p[b/v]$ and $q[b/v]$ are equal.

The first statement is clearly a condition. The second statement is translated into a condition as follows.

Let p be a process term. Applying rewrite rules 1-5 it can be reduced to a conditional term $\sum_i\{\alpha_i :\!\longrightarrow p_i\}$ in which all the monus signs have been replaced by minus signs. Now let $\sigma\in\Sigma$ and put

$$p\langle\sigma\rangle \;=\; \sum_{\{i\mid\sigma(\alpha_i)\text{ is true}\}} \sigma(p_i)$$

By applying rules 5,6 sufficiently many times we can reduce $p\langle\sigma\rangle$ to a process term in which all bounds are in normal form and redundant variables do not occur. This process term will be denoted by $\sigma(p)*$. Now for p, q process terms a condition $p\cong q$ is defined by

$$[p\cong q] \;:=\; \{\sigma\in\Sigma\mid\sigma(p)*\cong\sigma(q)*\}$$

It is easy to see that this collection is indeed a condition.

Let $p,q\in T$, $b\in Bound$ and $v\in TVar$. Ensure by applying α-conversion that $v\notin tvar(b)$ and that none of the variables occurring in b are bound by integrals occurring in p and q. Example 8.1 can be reduced by the following rewrite rule.

8. $\int_{v\in\langle b,c\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q \longrightarrow \{p[b/v]\cong q[b/v]\wedge b\in V :\!\longrightarrow \int_{v\in[b,c\rangle} a(v)\cdot p +$
$+\int_{v\in V} a(v)\cdot q\} \;+\; \{p[b/v]\ncong q[b/v]\vee b\notin V :\!\longrightarrow \int_{v\in\langle b,c\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q\}$

We also have a symmetric version of this rewrite rule in order to reduce the process term $\int_{v\in\langle c,b\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q$.

Similarly, Example 8.2 can be reduced by

9. $\int_{v \in V} a(v) \cdot p + \int_{v \in [b,b]} a(v) \cdot q \longrightarrow \{p[b/v] \cong q \wedge b \in V :\rightarrow \int_{v \in V} a(v) \cdot p\}$
$+ \{p[b/v] \not\cong q \vee b \notin V :\rightarrow \int_{v \in V} a(v) \cdot p + \int_{v \in [b,b]} a(v) \cdot q\}$

8.5 Constructing normal forms

In Paragraph 6.7 it has been proven that every process term can be reduced to a basic conditional term. We now show how to reduce a basic conditional term to a normal form, which will again be a basic conditional term.

First, let $\Sigma_i\{\alpha_i :\rightarrow p_i\}$ be a basic conditional term of depth 1. Then each p_i is of the form $\Sigma_j \int_{v \in V_{ij}} a_{ij}(v)$. Fix an i. We reduce p_i as follows.

- Apply rewrite rule 5 in order to reduce the bounds of the V_{ij} to normal form. Using rules 1-3 the conditions are reduced. Thus a basic conditional term is constructed of the form
$$\sum_k \{\beta_k :\rightarrow \sum_l \int_{v \in W_{kl}} b_{kl}(v)\}$$
where the bounds occurring in the W_{kl} and in the β_k are in normal form.

- Now apply rewrite rule 7a to each pair $\int_{v \in W_{kl}} b_{kl}(v) + \int_{v \in W_{kl'}} b_{kl'}(v)$ for which $b_{kl} = b_{kl'}$. Use rules 1-3 to reduce the conditions.

Thus we have constructed the normal form of p_i. Replace the p_i in $\Sigma_i\{\alpha_i :\rightarrow p_i\}$ by their normal forms. Use rule 1 to reduce conditions. The result is the normal form of $\Sigma_i\{\alpha_i :\rightarrow p_i\}$.

Now suppose that we have already constructed the normal forms for depth $\leq n$. Let $\Sigma_i\{\alpha_i :\rightarrow p_i\}$ be a basic conditional term of depth $n+1$. Fix an i and assume that
$$p_i \cong \sum_j \int_{v \in V_j} a_j(v) \cdot q_j + p'$$
where the q_j have depth n and p' has depth $\leq n$. According to the induction hypothesis we have already constructed normal forms for the q_j and for p'. We reduce $\Sigma_j \int_{v \in V_j} a_j(v) \cdot q_j$ as follows.

- Apply rewrite rule 5 in order to reduce the bounds of the V_j to normal form.

- Replace the q_j by their normal forms. Using rule 4, the conditions that occur in the normal form of q_j are lifted over the integral sign $\int_{v \in V_j}$. Apply rules 1-3 to reduce the conditions. Thus we have constructed a basic conditional term of the form
$$\sum_k \{\beta_k :\rightarrow \sum_l \int_{v \in W_{kl}} a_{kl}(v) \cdot q_{kl}\}$$

- Substitute redundant variables; if W_{kl} is of the form $[b,b]$ and $v \in FV(q_{kl})$, then apply rule 6 to $\int_{v\in W_{kl}} a_{kl}(v)\cdot q_{kl}$. Use rules 5,6 to reduce the bounds and to remove redundant variables in $q_{kl}[b/v]$.
- First apply rule 8 and then rules 7b and 9 to each pair $\int_{v\in W_{kl}} a_{kl}(v)\cdot q_{kl} + \int_{v\in W_{kl'}} a_{kl'}(v)\cdot q_{kl'}$ for which $a_{kl} = a_{kl'}$. Reduce the conditions using rules 1-3.

Now we have constructed the normal form of $\Sigma_j \int_{v\in V_j} a_j(v)\cdot q_j$. Add the normal form of p' to this term and reduce the conditions to a partition by applying rule 2. The result is the normal form of p_i.

Replace the p_i in $\Sigma_i\{\alpha_i :\!\rightarrow p_i\}$ by their normal forms. Use rule 1 to reduce the conditions. Thus we have constructed the normal form of $\Sigma_i\{\alpha_i :\!\rightarrow p_i\}$.

It is essential that in the reduction to normal form rewrite rule 8 is applied before rules 7b and 9 are. Consider for instance the term

$$\int_{v\in\langle 0,1\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w) + \int_{v\in\langle 1,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

from Example 8.3. First rule 8 is applied, resulting in

$$\int_{v\in\langle 0,1]} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w) + \int_{v\in[1,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

Then rules 7b and 9 are applied, giving the normal form $\int_{v\in\langle 0,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$.

9 Unique normal forms

Let $p \in \mathcal{T}^{cl}$ have normal form $\Sigma_i\{\alpha_i :\!\rightarrow p_i\}$. The construction of the normal form has been such that $tvar(\alpha_i) \subseteq FV(p)$, so each α_i is equal to either tt or ff. By applying the following two rewrite rules

10.	$\{tt :\!\rightarrow p\}$	$\longrightarrow$	p
11.	$\{ff :\!\rightarrow p\}$	$\longrightarrow$	δ

the normal form of p becomes a time-closed process term $p\!\downarrow$. We will prove that if $p, q \in \mathcal{T}^{cl}$ with $p = q$ (in BPA$\rho\delta I$), then $p\!\downarrow\ \cong q\!\downarrow$.

Lemma 9.1 *Let $p, q \in \mathcal{T}^{cl}$. If p can be rewritten to q by rules 1-11, then* BPA$\rho\delta I^- \vdash p = q$.

The proof of this lemma is identical to that of Proposition 6.3.

In the sequel 'normal form' will stand for normal form of a time-closed process term.

9.1 Two lemmas

Let b be a bound occurring in a normal form. Then b is of the form 0, ∞ or

$$(*) \quad (r_1\cdot v_1 + \ldots + r_k\cdot v_k + v_{k+1} + \ldots + v_l(+t)) - (s_1\cdot w_1 + \ldots + s_m\cdot w_m + w_{m+1} + \ldots + w_n(+t'))$$

where $v_1,\ldots,v_l,w_1,\ldots,w_n \in TVar$ are all different, $r_i, s_i \in \mathbb{Q}_{>0}\backslash\{1\}$ and either t, t' or both do not occur (see Appendix A).

Let b, c be bounds occurring in a normal form with $b \not\cong c$. Since b, c are of the form $(*)$, it follows that then there is at the most one $t \in V$ such that $b[t/v]* \cong c[t/v]*$. We now prove two lemmas.

A process term that is a subterm of a normal form is called a *subnormal form.*

Lemma 9.2 *Let p and q be subnormal forms. If $p[t/v]* \cong q[t/v]*$ for some $v \in TVar$ and infinitely many $t \in [0, \infty)$, then $p \cong q$.*

Proof. We use induction to the depth of p and q. Let

$$p \cong \sum_i \int_{w \in V_i} a_i(w) \cdot p_i + \sum_j \int_{w \in W_j} b_j(w)$$

$$q \cong \sum_k \int_{w \in V'_k} a'_k(w) \cdot q_k + \sum_l \int_{w \in W'_l} b'_l(w)$$

(In the first induction step the sums over i and k are empty). Assume that $p \not\cong q$. We distinguish two cases.

1. There is a j such that for all l we have $\int_{w \in W_j} b_j(w) \not\cong \int_{w \in W'_l} b'_l(w)$.

   Fix an l. If $b_j \neq b'_l$, then clearly $\int_{w \in W_j[t/v]*} b_j(w) \not\cong \int_{w \in W'_l[t/v]*} b'_l(w)$ for all t.

   So assume that $b_j = b'_l$. Then $W_j \not\cong W'_l$, so there is no more than one $t \in [0, \infty)$ such that $W_j[t/v]* \cong W'_l[t/v]*$.

   It follows that the set $\{t \in [0, \infty) \mid p[t/v]* \cong q[t/v]*\}$ 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 $\int_{w \in V_i} a_i(w) \cdot p_i \not\cong \int_{w \in V'_k} a'_k(w) \cdot q_k$.

   Fix a k. If $a_i \neq a'_k$ or $V_i \not\cong V'_k$, then it follows as in 1 that there is no more than one t such that

$$(\int_{w \in V_i} a_i(w) \cdot p_i)[t/v]* \cong (\int_{w \in V'_k} a'_k(w) \cdot q_k)[t/v]*$$

   So assume that $p_i \not\cong q_k$. Then by the induction hypothesis there is only a finite number of t such that $p_i[t/v]* \cong q_k[t/v]*$. Furthermore, if V_i resp. V'_k is not of the form $[b, b]$, then there is no more than one t such that $V_i[t/v]*$ resp. $V'_k[t/v]*$ does have this form.

   It follows that $\{t \in V \mid p[t/v]* \cong q[t/v]*\}$ is finite. □

Lemma 9.3 *Let $\int_{v \in V} a(v) \cdot p$ be a normal form. Then $p[t/v]*$ is a normal form for all $t \in V$.*

This lemma can be proven by showing that the construction described in Paragraph 8.5 reduces $p[t/v]*$ to itself. It is left to the reader to check that this is indeed the case.

9.2 Unique normal forms

Theorem 9.4 *Let $p, q \in \mathcal{T}^{cl}$. If $p = q$, then $p\downarrow \cong q\downarrow$.*

Proof. Assume that $p = q$. Lemma 9.1 implies $p = p\downarrow$ and $q = q\downarrow$, and so $p\downarrow = q\downarrow$. Then soundness gives $p\downarrow \leftrightarrow q\downarrow$. We prove that $p\downarrow \cong q\downarrow$, using induction to the depth of $p\downarrow$ and $q\downarrow$. First let

$$p\downarrow \cong \sum_{i\in I} \int_{v\in V_i} a_i(v) \qquad\qquad q\downarrow \cong \sum_{j\in J} \int_{v\in W_j} b_j(v)$$

Fix an $i \in I$. For $t \in V_i$ we have $p\downarrow \xrightarrow{a_i(t)} \surd$. Since $p\downarrow \leftrightarrow q\downarrow$, it follows that for each $t \in V_i$ there is a $j(t) \in J$ with $a_i = b_{j(t)}$ and $t \in W_{j(t)}$. Rewrite rule $7a$ has been applied, so then there must be a unique $j \in J$ with $a_i = b_j$ and $V_i \subseteq W_j$. Similarly for this j there is a unique $i(j) \in I$ with $b_j = a_{i(j)}$ and $W_j \subseteq V_{i(j)}$. Then rewrite rule $7a$ tells us that $i(j) = i$. Thus $V_i \equiv W_j$.

Now suppose that we have proven the theorem for depth $\leq n$. Let

$$p\downarrow \cong \sum_{i\in I} \int_{v\in V_i} a_i(v)\cdot p_i + p' \qquad\qquad q\downarrow \cong \sum_{j\in J} \int_{v\in W_j} b_j(v)\cdot q_j + q'$$

where the p_i and q_j have depth n and p' and q' have depth $\leq n$. Since $p\downarrow \leftrightarrow q\downarrow$, it follows that $p' \leftrightarrow q'$ and thus by the induction hypothesis $p' \cong q'$.

Fix an $i \in I$. Since $p\downarrow \leftrightarrow q\downarrow$, it follows that for each $t \in V_i$ there is a $j(t) \in J$ with $a_i = b_{j(t)}$ and $t \in W_{j(t)}$ and $p_i[t/v] \leftrightarrow q_{j(t)}[t/v]$.

1. First assume that V_i contains more than one point. Let $J' \subseteq J$ be the collection of j for which $b_j = a_i$ and $q_j \cong p_i$, and define $W_{J'} := \cup_{j\in J'} W_j$. Then $V_i \backslash W_{J'}$ is just a finite number of points.

   For suppose not. Clearly for each $t \in V_i \backslash W_{J'}$ we have $j(t) \notin J'$. So then there is an infinite subset S of $V_i \backslash W_{J'}$ and a $j_0 \in J \backslash J'$ such that $j(t) = j_0$ for all $t \in S$. Since $p_i[t/v] \leftrightarrow q_{j_0}[t/v]$, the induction hypothesis together with Lemma 9.3 tell us that $p_i[t/v]* \cong q_{j_0}[t/v]*$ for $t \in S$. Since S is infinite, Lemma 9.2 implies that $p_i \cong q_{j_0}$. It follows that $j_0 \in J'$, so we have a contradiction.

   Suppose that $t \in V_i \backslash W_{J'}$. Since this set is finite (and since V_i contains more than one point), there is a $j \in J'$ such that W_j is of the form $\langle t, t' \rangle$ or $\langle t', t \rangle$. The fact that $q_j \cong p_i$ together with $p_i[t/v] \leftrightarrow q_{j(t)}[t/v]$ implies $q_j[t/v] \leftrightarrow q_{j(t)}[t/v]$. Then the induction hypothesis together with Lemma 9.3 give $q_j[t/v]* \cong q_{j(t)}[t/v]*$. Since rewrite rule 8 has been applied to the pair

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

   we have $t \in W_j$. This is a contradiction, so it follows that $V_i \backslash W_{J'} = \emptyset$. This induces $V_i \subseteq W_{J'}$. Then rewrite rule $7b$ 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 $b_j = a_{i(j)}$ and $q_j \cong p_{i(j)}$ and $W_j \subseteq V_{i(j)}$. Rewrite rule $7b$ implies that $i(j) = i$. Thus $V_i \equiv W_j$.

2. Now assume that $V_i \equiv [t,t]$. If $W_{j(t)}$ contains more than one point, then we have just proven that there is an $i(t) \in I$ with $a_{i(t)} = b_{j(t)}$ and $V_{i(t)} \equiv W_{j(t)}$ and $p_{i(t)} \cong q_{j(t)}$. Since $V_{i(t)} \equiv W_{j(t)}$ it follows that $t \in V_{i(t)}$. And $p_{i(t)} \cong q_{j(t)}$ together with $p_i[t/v] \leftrightarrows q_{j(t)}[t/v]$ implies that $p_{i(t)}[t/v] \leftrightarrows p_i[t/v]$. Then the induction hypothesis together with Lemma 9.3 give $p_{i(t)}[t/v]* \cong p_i[t/v]*$. Since rewrite rule 9 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 $W_{j(t)} \equiv [t,t]$.

Since rewrite rule 6 has been applied, we have $p_i \cong p_i[t/v]*$ and $q_{j(t)} \cong q_{j(t)}[t/v]*$. The induction hypothesis together with Lemma 9.3 give $p_i[t/v]* \cong q_{j(t)}[t/v]*$, so $p_i \cong q_{j(t)}$. □

9.3 An example

The normal form of a process term can be much greater than the term itself. Consider for instance

$$\int_{v\in\langle 0,5\rangle} a(v)\cdot\Big(\int_{w\in\langle 7-4v,5-v\rangle} b(w) \;+\; \int_{w\in\langle 3,\frac{17}{6}+\frac{1}{3}v\rangle} b(w)\Big)$$

Its normal form can be deduced from Figure 4. 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:

$$\int_{v\in\langle 0,\frac{1}{2}]} a(v)\cdot\delta(v) \;+\; \int_{v\in\langle\frac{1}{2},\frac{2}{3}]} a(v)\cdot\int_{w\in\langle 3,\frac{17}{6}+\frac{1}{3}v\rangle} b(w)$$

$$+\; \int_{v\in\langle\frac{2}{3},\frac{25}{26}]} a(v)\cdot\Big(\int_{w\in\langle 3,\frac{17}{6}+\frac{1}{3}v\rangle} b(w) + \int_{w\in\langle 7-4v,5-v\rangle} b(w)\Big)$$

$$+\; \int_{v\in\langle\frac{25}{26},1]} a(v)\cdot\int_{w\in\langle 3,5-v\rangle} b(w) \;+\; \int_{v\in[1,\frac{7}{5}]} a(v)\cdot\int_{v\in\langle 7-4v,5-v\rangle} b(w)$$

$$+\; \int_{v\in[\frac{7}{5},\frac{13}{8}]} a(v)\cdot\int_{w\in\langle v,5-v\rangle} b(w) \;+\; \int_{v\in[\frac{13}{8},2\rangle} a(v)\cdot\int_{w\in\langle v,\frac{17}{6}+\frac{1}{3}v\rangle} b(w)$$

$$+\; \int_{v\in[2,\frac{5}{2}\rangle} a(v)\cdot\Big(\int_{w\in\langle v,5-v\rangle} b(w) + \int_{w\in\langle 3,\frac{17}{6}+\frac{1}{3}v\rangle} b(w)\Big)$$

$$+\; \int_{v\in[\frac{5}{2},3]} a(v)\cdot\int_{w\in\langle 3,\frac{17}{6}+\frac{1}{3}v\rangle} b(w) \;+\; \int_{v\in[3,\frac{17}{4}\rangle} a(v)\cdot\int_{w\in\langle v,\frac{17}{6}+\frac{1}{3}v\rangle} b(w)$$

$$+\; \int_{v\in[\frac{17}{4},5\rangle} a(v)\cdot\delta(v)$$

9.4 Relating BPA$\rho\delta I$ and BPA$\rho\delta I^-$

Extend the theory of CTA to CTA$^+$ by adding the axioms A1,2,4 of BPA and INT3a,b of BPA$\rho\delta I$ and rewrite rules 5-7.

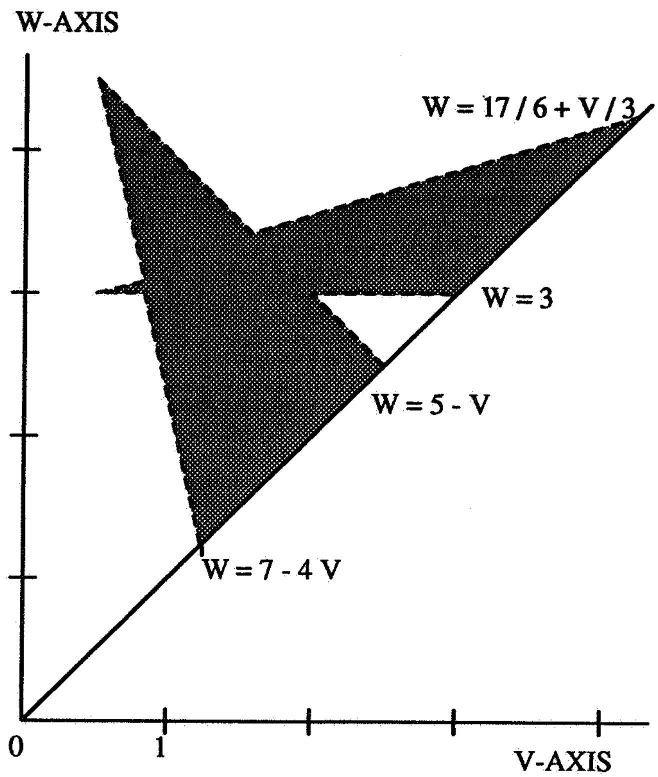

Figure 4: The graphical description of a process term

Proposition 9.5 $\forall p, q \in \mathcal{T}^{cl}$

$$\mathrm{BPA}\rho\delta I^- \vdash p = q \iff \mathrm{BPA}\rho\delta I \vdash p = q \iff \mathrm{CTA}^+ \vdash p = q$$

Proof. Clearly $\mathrm{BPA}\rho\delta I^- \vdash p = q$ implies $\mathrm{BPA}\rho\delta I \vdash p = q$

Now let $\mathrm{BPA}\rho\delta I \vdash p = q$. Then soundness implies $p \leftrightarrow q$. We can apply the reduction described in Paragraph 8.5 to reduce p and q to normal form. This reduction involves only axioms of CTA^+. Theorem 9.4 implies that $p\downarrow \cong q\downarrow$, so $\mathrm{CTA}^+ \vdash p = q$.

Finally, suppose that $\mathrm{CTA}^+ \vdash p = q$. Then it can be proven as in Proposition 6.3 that $\mathrm{BPA}\rho\delta I^- \vdash p = q$. □

A Reducing Bounds

The set of conditional bounds, with typical element b_c, is defined inductively as follows, where $b \in \mathit{Bound}$ and $\alpha \in \mathit{Cond}$.

$$b_c \ ::= \ b \mid \alpha :\rightarrow b_c \mid b_c + b_c \mid b_c \dot{-} b_c$$

Conditional bounds are considered modulo commutativity and associativity of the $+$.

We now show how to reduce a bound to a normal form, which will be a conditional bound of the form $\Sigma_i\{\alpha_i :\rightarrow b_i\}$. This construction is described in several steps, where each step consists of giving a collection of rewrite rules. Here, a rewrite rule is an expression of the form $b_c \longrightarrow b'_c$, where b_c, b'_c are conditional bounds.

Step 1

$$\begin{array}{|l|}\hline r\cdot(b+c) \longrightarrow r\cdot b + r\cdot c \\ r\cdot(b \dot{-} c) \longrightarrow r\cdot b \dot{-} r\cdot c \\ r\cdot t_0 \longrightarrow t_1 \text{ where } t_1 = r\cdot t_0 \\ r_0\cdot(r_1\cdot v) \longrightarrow r_2\cdot v \text{ where } r_2 = r_0\cdot r_1 \\ 1\cdot v \longrightarrow v \\ \hline\end{array}$$

Using these rules, each bound can be reduced to a bound that can be defined as follows, where $t \in \bar{D}$, $r \in D\backslash\{1\}$ and $v \in TVar$.

$$b ::= t \mid v \mid r\cdot v \mid b+c \mid b \dot{-} c$$

Step 2

Let $\{\alpha_i\}$ and $\{\beta_j\}$ be partitions and $\Box \in \{+, \dot{-}, -\}$. Then

$$\begin{array}{|l|}\hline b \dot{-} c \longrightarrow \{b > c :\longrightarrow b - c\} + \{b \le c :\longrightarrow 0\} \\ \Sigma_i\{\alpha_i :\longrightarrow b_i\} \Box \Sigma_j\{\beta_j :\longrightarrow c_j\} \longrightarrow \Sigma_{(i,j)}\{\alpha_i \wedge \beta_j :\longrightarrow b_i \Box c_j\} \\ \Sigma_i\{\alpha_i :\longrightarrow b_i\} \Box c \longrightarrow \Sigma_i\{\alpha_i :\longrightarrow b_i \Box c\} \\ b \Box \Sigma_j\{\beta_j :\longrightarrow c_j\} \longrightarrow \Sigma_j\{\beta_j :\longrightarrow b \Box c_j\} \\ \Sigma_i\{\alpha_i :\longrightarrow \Sigma_j\{\beta_j :\longrightarrow b_{ij}\}\} \longrightarrow \Sigma_{(i,j)}\{\alpha_i \wedge \beta_j :\longrightarrow b_{ij}\} \\ \hline\end{array}$$

The first rewrite rule is applied to each monus sign once. The minus sign can be considered to be the monus sign together with some index, telling that the first rewrite rule has been applied to this monus sign.

The bounds occurring in the conditions have to be reduced too. For instance, if $b \longrightarrow b'$ then $b \le c \longrightarrow b' \le c$.

With these rewrite rules each bound can be reduced to the form $\Sigma_i\{\alpha_i :\longrightarrow b_i\}$, where the bounds b_i and the bounds occurring in the α_i do not contain monus signs.

Step 3

$$\begin{array}{|l|}\hline b_0 + (b_1 - b_2) \longrightarrow (b_0 + b_1) - b_2 \\ (b_0 - b_1) + b_2 \longrightarrow (b_0 + b_2) - b_1 \\ b_0 - (b_1 - b_2) \longrightarrow (b_0 + b_2) - b_1 \\ (b_0 - b_1) - b_2 \longrightarrow b_0 - (b_1 + b_2) \\ \hline\end{array}$$

Using these rewrite rules, each bound can be reduced to the form $\Sigma_i\{\alpha_i :\longrightarrow b_i\}$, where the bounds b_i and the bounds that occur in the α_i are of the form

$$(r_1 \cdot v_1 + ... + r_i \cdot v_i + v_{i+1} + ... + v_j + t_1 + ... + t_k) - (s_1 \cdot w_1 + ... + s_l \cdot w_l + w_{l+1} + ... + w_m + t'_1 + ... + t'_n)$$

with $j + k \geq 1$ and $m + n \geq 0$.

Step 4

Let $\Box \in \{+, -\}$.

$t_0 + t_1 \longrightarrow t_2$ where $t_2 = t_0 + t_1$
$r_0 \cdot v + r_1 \cdot v \longrightarrow r_2 \cdot v$ where $r_2 = r_0 + r_1$
$r_0 \cdot v + v \longrightarrow r_1 \cdot v$ where $r_1 = r_0 + 1$
$v + v \longrightarrow 2 \cdot v$
$b \Box 0 \longrightarrow b$
$\infty \Box b \longrightarrow \infty$

With these rewrite rules we can reduce each bound to the form $\Sigma_i\{\alpha_i :\longrightarrow b_i\}$, where the bounds b_i and the bounds that occur in the α_i are of the form 0, ∞ or

$$(*) \quad (r_1 \cdot v_1 + ... + r_i \cdot v_i + v_{i+1} + ... + v_j(+t)) - (s_1 \cdot w_1 + ... + s_l \cdot w_l + w_{l+1} + ... + w_m(+t'))$$

where $v_1, ..., v_j \in TVar$ are all different, $w_1, ..., w_m \in TVar$ are all different and $t, t' \in D$ are not necessarily there (that is why they have been put between brackets).

Step 5

Finally, by applying rewrite rules like

$(b + r_0 \cdot v) - (c + r_1 \cdot v) \longrightarrow (b + r_2 \cdot v) - c$ if $r_0 > r_1$ and $r_2 = r_0 - r_1$
$(b + r_0 \cdot v) - (c + r_1 \cdot v) \longrightarrow b - (c + r_2 \cdot v)$ if $r_0 < r_1$ and $r_2 = r_1 - r_0$
$(b + r \cdot v) - (c + r \cdot v) \longrightarrow b - c$

we can reduce bounds of the form $(*)$ such that $v_1, ..., v_j, w_1, ..., w_m \in TVar$ are all different and either t, t' or both do not occur.

References

[BB90] J.C.M. Baeten and J.A. Bergstra. Process algebra with signals and conditions. Report P9008, University of Amsterdam, Amsterdam, 1990.

[BB91] J.C.M. Baeten and J.A. Bergstra. Real time process algebra. *Journal of Formal Aspects of Computing Science*, 3(2):142–188, 1991.

[BK84] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Computation*, 60(1/3):109–137, 1984.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra*. Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.

[Fok91] W.J. Fokkink. Normal forms in real time process algebra. Report CS-R9149, CWI, Amsterdam, 1991.

[Gla87] R.J. van Glabbeek. Bounded nondeterminism and the approximation induction principle in process algebra. In F.J. Brandenburg, G. Vidal-Naquet, and M. Wirsing, editors, *Proceedings STACS 87*, volume 247 of *Lecture Notes in Computer Science*, pages 336–347. Springer-Verlag, 1987.

[Gro89] J.F. Groote. Transition system specifications with negative premises. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90*, Amsterdam, LNCS 458, pages 332–341. Springer-Verlag, 1990.

[Gro90] J.F. Groote. Specification and verification of real time systems in ACP. In L. Logrippo, R.L. Probert and H. Ural, editors, *Proceedings* 10^{th} *International Symposium on Protocol Specification, Testing and Verification*, Ottawa, pages 261–274, 1990.

[Klu91] A.S. Klusener. Completeness in real time process algebra. Report CS-R9106, CWI, Amsterdam, 1991. An extended abstract appeared in J.C.M. Baeten and J.F. Groote, editors, *Proceedings CONCUR 91*, Amsterdam, LNCS 527 , pages 376–392. Springer-Verlag, 1991.

[Mil89] R. Milner. *Communication and concurrency.* Prentice Hall International, 1989.

[MT90] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*, pages 401–415. Springer-Verlag, 1990.

[NS90] X. Nicollin and J. Sifakis. ATP: An algebra for timed processes. Technical Report RT-C26, IMAG, Laboratoire de Génie informatique, Grenoble, 1990.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, 5^{th} *GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[Plo81] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[RR88] M. Reed and A.W. Roscoe. A timed model for communicating sequential processes. *Theoretical Computer Science*, 58:249–261, 1988.

[Wan90] Y. Wang. Real time behaviour of asynchronous agents. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*, pages 502–520. Springer-Verlag, 1990.