1991

A.S. Klusener

Completeness in real time process algebra

Computer Science/Department of Software Technology Report CS-R9106 January

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

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).

Completeness in Real Time Process Algebra

A.S. Klusener

CWI
P.O. Box 4079, 1009 AB Amsterdam,
The Netherlands
e-mail: stevenk@cwi.nl

Abstract

Recently, J.C.M. Baeten and J.A. Bergstra extended ACP with real time, resulting in a Real Time Process Algebra, called ACPρ [BB89]. They introduced an equational theory and an operational semantics. Their paper does not contain a completeness result nor does it contain the definitions to give proofs in detail. In this paper we introduce some machinery and a completeness result.

The operational semantics of [BB89] contains the notion of an *idle* step reflecting that a process can do nothing more then passing the time before performing a concrete action at a certain point in time. This *idle* step corresponds nicely to our intuition but it results in uncountable branching transition systems. In this paper we give a more abstract operational semantics, by abstracting from the *idle* step. Due to this simplification we can prove soundness and completeness easily. These results hold for the semantics of [BB89] as well, since both operational semantics induce the same equivalence relation between processes.

1985 Mathematics Subject Classification: 68Q10, 68Q40, 68Q45, 68Q55.
1982 CR Categories: D.1.3, D.3.1, D.4.1, F.1.2, F.3.2.
Key Words & Phrases: Real Time, Process Algebra, ACP, Integration, SOS.
Note: This work is in part sponsored by ESPRIT Basic Research Action 3006, CONCUR.

Introduction

In order to reason algebraically with concurrent software systems, e.g. communication protocols and their specifications, process algebras such as CCS ([Mil80]), CSP ([Hoa85]) and ACP ([BK84],[BW90]) are developed. Since some communication protocols contain timed constructs, such as timers and time-outs, there is a need to extend these process algebras with a notion of real time. Several authors have given a real time extension of a process algebra, e.g. see [MT90],[Wan90] for an extension of CCS, see [RR88] for an extension of CSP and see [BB89], [Gro90] for an extension of ACP. Nicollin and Sifakis have given a real time extension of a combination of CCS and ACP in [NS90]. Between these extensions several similarities and differences can be found. For example, most of these extensions are based on discrete time. However, the paper of Baeten and Bergstra is based on dense time and it introduces the interesting notion of integration, by which we can express that an action occurs somewhere within a (dense) interval. Hence, their work is more general and more complicated. In general their theory is undecidable, due to the notion of general integration over arbitrary subsets of $\mathbb{R}^{\geq 0}$. In this report we restrict ourselves to *prefixed* integration to tackle the complexity of integration over sets of reals. *Prefixed* integration requires that every action, which has a *time variable* v as timestamp, is directly preceded by the binding integral of v. This restriction seems to be harmless from a practical point of view, since no "real-life" processes are known to us which are not covered by this restriction. Due to this simplification we can obtain a completeness result.

Real Time Process Algebra ([BB89]) concerns terms constructed from timed actions. A timed action is a combination of a symbolic action, taken from some alphabet A, and a timestamp, taken from $\mathbb{R}^{\geq 0}$. This timestamp can be interpreted either absolute or relative. In absolute time every timestamp is

Report CS-R9106
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

interpreted from the starttime of the entire process (time zero) while in relative time every timestamp is interpreted from the point in time of the execution of the previous action. In this paper we will focus on absolute time only, in [BB89] a translation from relative to absolute time is given. Thus, the timed action $a(2)$ denotes the atomic process which executes an action a at time 2, after which it terminates successfully. We expect some new 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 and in the following alternative composition $b(1) + c(3)$ the first alternative cannot be chosen anymore and therefore it may be removed.

Furthermore, we consider integration as an essential feature of our theory. Integration is the alternative composition over a continuum of alternatives ([BB89]). An integral describes the interval in which a timed action may occur. For example, the process which executes an action a somewhere within the interval $< 0, 1 >$ is denoted by $\int_{v \in <0,1>} a(v)$.

According to [BB89] we refer to Basic Real Time Process Algebra (the theory without parallelism) as BPA$\rho\delta$, by adding integration we get BPA$\rho\delta$I. Moreover, to Real Time Process Algebra (the theory with parallelism) is referred to as ACPρresp. ACPρI.

The development of such a theory implies the development of a system of axioms which generates exactly the required identities. To be sure that this system of axioms is exactly what we have in mind, we give an operational semantics. In our case an operational semantics assigns to every process expression a transition system which reflects the behaviour of this process. A transition system is a directed acyclic graph in which each arrow is labelled with a timed action. These transitions systems are generated by giving a Transition System Specification in Structured Operational Semantics according to Plotkin ([Plo81]). We define an equivalence notion (which must be a congruence) on these transition systems by defining a bisimulation relation ([Mil89],[Par81]). If we have defined such an operational semantics we can validate the axiom system by proving soundness and completeness.

In [BB89] an axiom system is given together with an operational semantics which yields uncountable transition systems even for terms without integration and recursion. In that paper, neither soundness nor completeness are treated thoroughly.

In this paper we will give an abstract operational semantics which yields finite transition systems for terms without integration and recursion. The action rules defining this abstract semantics are similar to the action rules for the standard operational semantics of ACP as given in [Gla87]. We prove soundness and completeness of this abstract operational semantics with respect to Real Time Process Algebra. We can extend these results to the semantics of [BB89] since we prove that both semantics are equivalent in the sense that they induce the same identities between terms.

In this paper we define the notion of a *basic* term. A *basic* term has very nice properties, like a strong correspondence between a *basic* term an its transition system. Hence, we can prove completeness easily for this subset of terms. Since we prove as well that every term is derivably equal to a *basic* term within the theory, which is sound, we can extend the completeness result to all transition systems which can be represented by a term.

We do not consider recursion in this paper.

Contents

1 The Original Operational Semantics

In this section we give some intuition for timed processes by introducing the operational semantics of [BB89] for process expressions over Basic Real Time Process Algebra (BPA$\rho\delta$). We do not yet consider integration in this section. Let A_δ be the set of actions, containing the constant δ. The alphabet of the theory BPA$\rho\delta$ is

$$A_\delta^{time} = \{a(t) | a \in A_\delta, t \in \mathbb{R}^{\geq 0}\}$$

Similarly we use A^{time}, as the set of timed actions without timed δ's. In the sequel we refer to actions from A_δ as symbolic actions and we refer to actions from A_δ^{time} as timed actions. Moreover, process expressions are simply called terms. The set $\mathcal{T}$ of closed terms over BPA$\rho\delta$ is generated by the alphabet A_δ^{time} and the binary operators $+$ for alternative composition and $\cdot$ for sequential composition and the operator $\gg$, called the *(absolute) time shift*. In the previous sentence we referred to closed terms as terms without recursion variables. In this report we do not consider recursion, hence if we consider a term it is meant to be a term which is closed w.r.t. recursion variables. In the sequel we will introduce another kind of variables, called *time variables* .

The *(absolute) time shift*, $\gg$, takes a nonnegative real number and process term; $t \gg X$ denotes that part of X, which starts after t. The set $\mathcal{T}$ with typical elements p, p_1, p_2 is defined in the following way, where $a \in A_\delta$ and $r \in \mathbb{R}^{\geq 0}$:

$$p \in \mathcal{T} \quad p ::= a(r) \mid p_1 + p_2 \mid p_1 \cdot p_2 \mid r \gg p$$

The semantics of [BB89] assigns to every term (in $\mathcal{T}$) a transition system in which each state is a pair consisting of a term and a point in time and in which each transition is labelled by a timed (non δ) action. Within this semantics each transition system concerns three relations

$$\begin{array}{lll} \mathit{Step} & \subseteq & (\mathcal{T} \times \mathbb{R}^{\geq 0}) \times A^{time} \times (\mathcal{T} \times \mathbb{R}^{\geq 0}) \\ \mathit{Idle} & \subseteq & (\mathcal{T} \times \mathbb{R}^{\geq 0}) \times (\mathcal{T} \times \mathbb{R}^{\geq 0}) \\ \mathit{Terminate} & \subseteq & (\mathcal{T} \times \mathbb{R}^{\geq 0}) \times A^{time} \times \mathbb{R}^{\geq 0} \end{array}$$

These three relations are defined as the least relations satisfying the Plotkin 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*. The term $a(1)$ denotes the process which performs an action at time 1, after which it is successfully terminated. The root node of the transition system of $a(1)$ is $< a(1), 0 >$, denoting that time starts at zero. In Table 1 the action rules for timed actions and the alternative composition are given (a is taken from A).

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

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

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

Table 1: Action Rules for Atomic Actions and Alternative Composition

From $< a(1), 0 >$ an *idle* transition is possible to a state of the form $< a(1), t >$ with $0 < t < 1$. An *idle* transition is a transition in which only the time component is increased without executing any action. In general, from each state $< a(1), t >$ an *idle* transition is possible to $< a(1), t' >$, whenever $t < t' < 1$. Further from each state $< a(1), t >$ a $a(1)$-transition to $< \surd, 1 >$ is possible whenever $t < 1$.

This 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 in 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

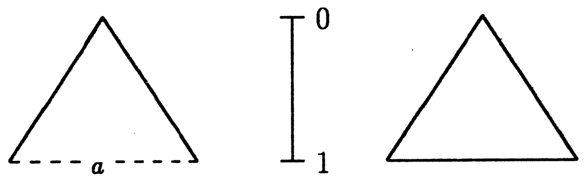

Figure 1: Process Diagrams of the terms $a(1)$ and $\delta(1)$

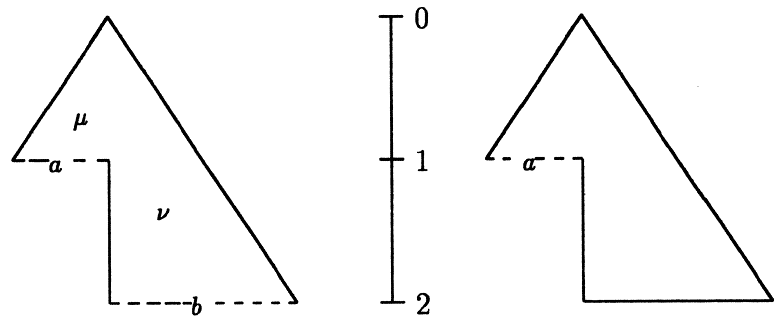

Figure 2: Process Diagrams of the terms $a(1) + b(2)$ and $a(1) + \delta(2)$

reflected by going to a dashed line at level r labelled with a. Only dashed lines may be crossed, after landing on them.

A very particular set of atomic actions is the set of $\delta(r)$-terms. $\delta(1)$ can do nothing more then idling until 1. Thus the root node is $< \delta(1), 0 >$ and from each state $< \delta(1), t >$ an *idle* transition to $< \delta(1), t' >$ is possible, whenever $t < t' < 1$. The action rules defining the transition systems of timed actions are given in Table 1. In Table 1 the rules for the alternative composition are given as well; the transition system of $p + q$ is defined in terms of the transition systems of p and q. The behaviour of $p + q$ can be considered as the "union" of the behaviour of p and that of q.

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 $0 < t < 1$. 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 have lost the possibility of executing the $a(1)$-summand. Thus one could say that a choice has been made at time 1; after the choice has been made for $b(2)$ the summand $a(1)$ has become 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* steps which are contributed by the summand $a(1)$ as well.

However if we consider $a(1) + \delta(2)$, 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 in Figure 2. The action rules for sequential composition, are given in Table 1.

The last operator we introduce is the *(absolute) time shift* denoted by $\gg$, which takes a real number s and a process X and delivers that part of X which starts after s. Hence, before s it can only *idle* or do a transition to a state after s. The action rules for the *(absolute) time shift* operator are

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

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

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

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

Table 2: Action Rules for Sequential Composition And Bound Shift

given in Table 1 as well. Note that we can show by induction of the length of the derivation that $<x,t> \xrightarrow{a(r)} <x',t'>$ implies that $t<r$ and $t'=r$. Or, the execution of an action a at time r is possible only in a state with a time component smaller than r.

2 The Theory of Basic Real Time Process Algebra

2.1 BPA$\rho\delta$

BPA$\rho\delta$ is the theory of Basic Real Time Process Algebra ([BB89]). It consists of $A1 - A5$, which are the standard axioms of Basic Process Algebra, extended by some axioms stating the specific real time properties and defining the *(absolute) time shift.* The axioms of BPA$\rho\delta$ are given in Table 3, we abbreviate $\delta(0)$ by δ. Notice that the laws ATA2 and ATA4 are generalisations of the BPA$_\delta$ laws A6 and A7.

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

$$\begin{array}{lcl}
5 \gg (a(4)+b(6)+c(7)\cdot d(8)) & = & b(6)+c(7)\cdot d(8) \\
5 \gg (a(4)+b(3)) & = & \delta(5) \\
\delta(1)+a(2)\cdot b(3)+\delta(3)\cdot c(4) & = & a(2)\cdot b(3)+\delta(3) \\
a(0)+b(2)\cdot(c(1)+c(3))+d(3)\cdot e(2) & = & b(2)\cdot c(3)+d(3)\cdot\delta(3)
\end{array}$$

Some Definitions and Properties

We introduce some definitions of inductive functions on terms; $U(p)$ is the *ultimate delay* which is the smallest timestamp which is not reachable by the process p by idling. *inittim*(p) is the set of points in time at which p can execute an initial action. $L(p)$ is the latest moment at which p can execute an action.

Definition 2.1.1 *Definition of the syntactic operations U(p), inittim(p), L(p).*

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		$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$
		BPA = A1 − A5
$a \in A_\delta$		
ATA1		$a(0) = \delta$
ATA2		$\delta(t) \cdot X = \delta(t)$
ATA3	$t \leq r$	$\delta(t) + \delta(r) = \delta(r)$
ATA4		$a(t) + \delta(t) = a(t)$
ATA5		$a(t) \cdot X = a(t) \cdot (t \gg X)$
ATB1	$t < r$	$t \gg a(r) = a(r)$
ATB2	$t \geq r$	$t \gg a(r) = \delta(t)$
ATB3		$t \gg (X + Y) = (t \gg X) + (t \gg Y)$
ATB4		$t \gg (X \cdot Y) = (t \gg X) \cdot Y$

Table 3: BPA$\rho\delta$ =BPA+ATA1-5+ATB1-4

$a \in A$

$$
\begin{aligned}
U(\delta(t)) &= t \\
U(a(t)) &= t \\
U(X \cdot Y) &= U(X) \\
U(X + Y) &= max(U(X), U(Y)) \\
U(r \gg X) &= max(U(X), r) \\
\\
inittim(\delta(t)) &= \emptyset \\
inittim(a(t)) &= \begin{cases} \emptyset & \text{if } t = 0 \\ \{t\} & \text{otherwise} \end{cases} \\
inittim(X \cdot Y) &= inittim(X) \\
inittim(X + Y) &= inittim(X) \cup inittim(Y) \\
inittim(r \gg X) &= \{t \mid t \in inittim(X) \ \wedge \ t > r\} \\
\\
L(X) &= max(inittim(X)) \text{ where } max(\emptyset) = 0
\end{aligned}
$$

The *ultimate delay* is introduced already in [BB89], however there it is not defined as a function on the syntax but as an operator within the theory. The other functions are new here. The *size* of a term X is defined as the number of operators in the term X.

Terms can be considered equal in different ways. Therefore we need some different notations. Two terms p, q are syntacticly 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 syntacticly equivalent, but more whether they are equal modulo the axioms A1 and A2, denoted by $p \simeq q$. Sometimes we refer to this equivalence as "the terms 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 they are called derivably equal to each other, denoted by $p = q$. Finally, we have two notions of summand inclusion, the idea is that p is a summand of $p + q$. We use *derivable summand inclusion*, if we consider the terms in the context of the theory. However, if we want to consider the form of the terms only, for example when we compare different semantics, we need a notion of *syntactic summand inclusion.*

Definition 2.1.2

syntactic equivalence		$p \equiv q$
syntactic equivalence modulo A1 *and* A2	A1,A2 $\vdash p = q$	$p \simeq q$
derivability in BPA$\rho\delta$	BPA$\rho\delta \vdash p = q$	$p = q$
derivable summand inclusion	BPA$\rho\delta \vdash p + q = q$	$p \subseteq q$
syntactic summand inclusion	*see below*	$p \sqsubseteq q$

$\sqsubseteq$ *is the least relation satisfying:*

$$\begin{array}{lcl} p \simeq q & \Longrightarrow & p \sqsubseteq q \\ p \sqsubseteq q & \Longrightarrow & p \sqsubseteq q + q' \end{array}$$

For convenience we generalise the definition of $\simeq$ by adding $p \simeq p + \delta$ and taking the transitive closure. Note that δ is an abbreviation of $\delta(0)$.

Now we can state:

$$\begin{array}{lcl} U(p) \leq t & \Longleftrightarrow & a(t) \cdot p = a(t) \cdot \delta(t) \\ p = q & \Longrightarrow & active(p) = active(q) \ \wedge \ inittim(p) = inittim(q) \\ U(t \gg p) & = & max(t, U(p)) \end{array}$$

2.2 Basic Terms

If we consider terms like $a(5) \cdot b(6) + \delta(1)$ and $a(5) \cdot (b(6) + c(4))$, we may say that both of them contain "redundant" parts; parts that can be removed by application of the axioms of BPA$\rho\delta$. Within BPA$\rho\delta$, both terms are derivably equal to the term $a(5) \cdot b(6)$, which does not contain any redundant parts. Hence, this last term can be seen as a kind of normal form and in the sequel we will call it a *basic* term.

$\mathcal{B}$ is the set of *basic* terms. To define this set of *basic* terms we first define auxiliary sets $\mathcal{B}^{\,r}(t)$ with $r \in \mathbb{R}^{\geq 0}_{\omega}$ and $t \in \mathbb{R}^{\geq 0}$, where $\mathbb{R}^{\geq 0}_{\omega} = \mathbb{R}^{\geq 0} \cup \{\omega\}$. We extend the standard ordering $<$ on $\mathbb{R}^{\geq 0}$ to an ordering on $\mathbb{R}^{\geq 0}_{\omega}$ by putting $\forall s \in \mathbb{R}^{\geq 0} : s < \omega$. $p \in \mathcal{B}^{\,r}(t)$ means that p is a *basic* term which can *idle* until t without losing alternatives and it has a deadlock at r whenever $r \neq \omega$. Hence, we have $r > t$ in case $\mathcal{B}^{\,r}(t) \neq \emptyset$. Moreover, $\mathcal{B}\,(t)$ is the set of *basic* terms starting after t, regardless of a possible deadlock.

Definition 2.2.1 *Let* $p, q \in \mathcal{T}, a \in A, t \in \mathbb{R}^{\geq 0}$ *and* $r, s \in \mathbb{R}^{\geq 0}_{\omega}$.

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

$$\begin{array}{rlcl} t < s & a(s) & \in & \mathcal{B}^{\,\omega}(t) \\ t < s & \delta(s) & \in & \mathcal{B}^{\,s}(t) \\ t < s \ \wedge \ (p \in \mathcal{B}\,(s)) \ \vee \ p \simeq \delta) & a(s) \cdot p & \in & \mathcal{B}^{\,\omega}(t) \\ p \in \mathcal{B}^{\,r}(t) \ \wedge \ q \in \mathcal{B}^{\,z}(t) \ \wedge \ (\, r = z \ \vee \ (r < z = \omega \ \wedge \ r > U(q))\,) & p + q, q + p & \in & \mathcal{B}^{\,r}(t) \\ p \in \mathcal{B}^{\,r}(t) & p + \delta, \delta + p & \in & \mathcal{B}^{\,r}(t) \\ & & & \\ & \mathcal{B}\,(t) & = & \cup_r \mathcal{B}^{\,r}(t) \\ & \mathcal{B} & = & \mathcal{B}\,(0) \cup \{\delta\} \end{array}$$

Note that *basic* terms have the following properties

Lemma 2.2.2

$$\begin{array}{lcl} t \le t' & \Longleftrightarrow & \mathcal{B}(t) \supseteq \mathcal{B}(t') \\ a(t)\cdot p \in \mathcal{B} \;\wedge\; p \not\simeq \delta & \Longleftrightarrow & p \in \mathcal{B}(t) \\ \delta(t) \sqsubseteq p \;\wedge\; t > 0 & \Longleftrightarrow & t = U(p) > L(p) \end{array}$$

Proof. Direct by the definition of *basic* terms □

By definition of *basic* terms we know that only prefix multiplication is used, and the *(absolute) time shift* operator $\gg$ does not occur in a basic term. In the sequel we abbreviate terms of the form $p_1 + ... + p_n$ by $\sum_{1\le i\le n} p_i$ or $\sum_{i\in\{1,...,n\}} p_i$. As is usual we use the convention $\sum_{i\in\emptyset} p_i \simeq \delta$. Hence, every *basic* term is of the form

$$\sum_{1\le i\le n} a_i(t_i)\cdot p_i \;+\; \sum_{1\le j\le m} b_j(r_j)$$

where $n+m>0$, $p_i \in \mathcal{B}$ and a_i and b_j are atomic actions. The addition of terms like $p+\delta$ to $\mathcal{B}$ has technical reasons. Due to this addition we can have empty summations within a *basic* term. Moreover, by the previous Lemma we know:

$$\begin{array}{l} \forall i \quad a_i \ne \delta \;\wedge\; (p_i \in \mathcal{B}(t_i) \;\vee\; p_i \simeq \delta) \\ U(p) > L(p) \quad \Longleftrightarrow \quad \exists j \quad b_j = \delta \;\wedge\; r_j = U(p) \end{array}$$

Examples of *basic* terms:

$$\begin{array}{llcl} \forall t < 5 & b(5) & \in & \mathcal{B}^{\omega}(t) \\ \forall r < 10 & \delta(10) & \in & \mathcal{B}^{10}(r) \\ \forall s < 5 & b(5)+\delta(10) & \in & \mathcal{B}^{10}(s) \\ \forall u < 2 & a(2)\cdot(b(5)+\delta(10)) & \in & \mathcal{B}^{\omega}(u) \end{array}$$

Lemma 2.2.3 *For each* $p, q \in \mathcal{B}$ *there is a* $z \in \mathcal{B}$ *such that* $\mathrm{BPA}\rho\delta \vdash z = p\cdot q$

Proof. Assume

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

We take

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

Construct $\overline{p}$ from p by removing the $\delta(t)$ summands, hence

$$\begin{array}{lcl} \overline{J} & = & \{j \in J \mid b_j \ne \delta\} \\ \overline{p} & \simeq & \sum_{i\in I} a_i(r_i)\cdot p_i + \sum_{j\in\overline{J}} b_j(s_j) \end{array}$$

Now, we can proof the Lemma by induction on the depth of p.

- $I=\emptyset$ and $J \ne \emptyset$
  Construct first $\overline{z}$ such that $\overline{z} = \overline{p}\cdot q = \sum_{j\in\overline{J}} b_j(s_j)\cdot(s_j \gg q)$ by taking

$$\overline{z} \simeq \sum_{j\in\overline{J}} b_j(s_j)\cdot\{\sum_{k\in K^{s_j}} c_k(t_k)\cdot r_k + \sum_{l\in L^{s_j}} d_l(u_l)\}$$

  Then we can construct z as follows:

$$z \simeq \begin{cases} \overline{z} + U(p) \in \mathcal{B} & \text{if } U(p) > U(\overline{p}) \\ \overline{z} \in \mathcal{B} & \text{otherwise} \end{cases}$$

- $I \neq \emptyset$

$$\begin{array}{rcl} \overline{p} \cdot q & \simeq & (\sum_{i \in I} a_i(r_i) \cdot p_i + \sum_{j \in \overline{J}} b_j(s_j)) \cdot q \\ & = & \sum_{i \in I}(a_i(r_i) \cdot p_i) \cdot q + \sum_{j \in \overline{J}} b_j(s_j) \cdot q \\ & = & \sum_{i \in I} a_i(r_i) \cdot (r_i \gg (p_i \cdot q)) + \sum_{j \in \overline{J}} b_j(s_j) \cdot (s_j \gg q) \end{array}$$

Since $p_i \in \mathcal{B}\,(r_i)$ we know $r_i \gg (p_i \cdot q) = (r_i \gg p_i) \cdot q = p_i \cdot q$.
By induction there is for each $i \in I$ a $z_i \in \mathcal{B}$ such that $z_i = q_i \cdot q$.
Hence, by constructing $\overline{z}$ as follows we have $\overline{z} = \overline{p} \cdot q$

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

And we can construct $z \in \mathcal{B}$ satisfying $z = p \cdot q$ by taking

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

□

Lemma 2.2.4 *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 $t = 0$
   Then
   $$p \equiv a(0) \overset{\text{ATA1}}{=} \delta \in \mathcal{B}$$
   Else
   $$p \equiv a(t) \in \mathcal{B}$$

2. $p \equiv q \cdot r$
   By induction there are $q_b, r_b \in \mathcal{B}$ such that BPA$\rho\delta \vdash q = q_b$ and BPA$\rho\delta \vdash r = r_b$ and the case follows by the previous Lemma.

3. $p \equiv q + r$
   By induction there are $q_b, r_b \in \mathcal{B}$ such that BPA$\rho\delta \vdash q = q_b$ and BPA$\rho\delta \vdash r = r_b$, say
   $$\begin{array}{rcl} q_b & \simeq & \sum_{i \in I} a_i(r_i) \cdot q_i + \sum_{j \in J} b_j(s_j) \\ r_b & \simeq & \sum_{k \in K} c_k(t_k) \cdot r_k + \sum_{l \in L} d_l(u_l) \end{array}$$
   Construct $\overline{q}_b$ and $\overline{r}_b$ from q_b and r_b by removing the $\delta(t)$ summands, hence
   $$\begin{array}{rcl} \overline{J} & = & \{j \in J \mid b_j \neq \delta\} \\ \overline{L} & = & \{l \in L \mid d_l \neq \delta\} \\ \overline{q}_b & \simeq & \sum_{i \in I} a_i(r_i) \cdot q_i + \sum_{j \in \overline{J}} b_j(s_j) \\ \overline{r}_b & \simeq & \sum_{k \in K} c_k(t_k) \cdot r_k + \sum_{l \in \overline{L}} d_l(u_l) \end{array}$$
   And we can conclude
   $$p = \begin{cases} \overline{q}_b + \overline{r}_b + \delta(U(q_b + r_b)) \in \mathcal{B} & \text{if } U(q_b + r_b) > U(\overline{q}_b + \overline{r}_b) \\ \overline{q}_b + \overline{r}_b \in \mathcal{B} & \text{otherwise} \end{cases}$$

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

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

   Define I^r and J^r as

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

   Now

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

□

3 An Abstract Operational Semantics for BPA$\rho\delta$

3.1 An Abstraction from the *idle* transition

In the first section the operational semantics of [BB89] is presented. In this operational semantics every transition system has uncountably many *idle* transitions. In this section we abstract from these *idle* transitions; we redefine the operational semantics such that there are no *idle* transitions anymore and hence all transition systems of finite closed BPA$\rho\delta$ terms have a finite number of transitions.

The transition system of $a(1)$ will now contain only one labelled arrow as given in the left-hand side of Figure 3. This new semantics is analogous to the semantics for *ACP* presented in [Gla87], where the transition system of a is simply:

$$a \xrightarrow{a} \surd$$

Moreover, in the same semantics the transition system of $a \cdot p$ contains the transition

$$a \cdot p \xrightarrow{a} p$$

In a real time setting we have to take the time stamps into account. In $a(r) \cdot p$, after doing the $a(r)$ action, only that part of p can be done which starts after r, which is denoted by $r \gg p$. Hence, the transition system of $a(r) \cdot p$ contains the transition

$$a(r) \cdot p \xrightarrow{a(r)} r \gg p$$

If p can perform an action $b(t)$ then $s \gg p$ can perform this action only when $t > s$.

In Figure 3 the "abstract" transition systems of the terms $a(1)$ and $a(1) + b(2)$ are given, together with the corresponding process diagrams. We have to be careful with deadlocks, because we have to distinguish the term $\delta(1)$ from $\delta(2)$. In the next section we add an extra condition to the definition of bisimulation, saying that only nodes with the same *ultimate delay* are bisimilar. Since states are now terms from $\mathcal{T}$, every state has an *ultimate delay*. In a later section we will introduce a relation *Deadlock* $\subseteq \mathcal{T} \times \mathbb{R}^{\geq 0}$, like the relations *Step* and *Terminate*, where $(p, r) \in$ *Deadlock* means that p has a deadlock at r.

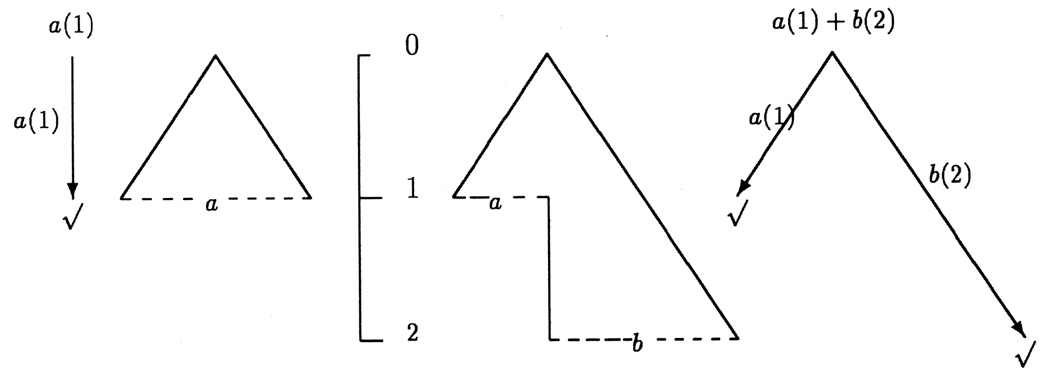

Figure 3: Process Diagrams and Transition Systems without *idle* steps for the terms $a(1)$ resp. $a(1)+b(2)$

3.2 An Abstract Semantics with an Extended Notion of Bisimulation

Now we can give the more formal definition in Structured Operational Semantics (see Table 4). In this abstract semantics every state is a term from $\mathcal{T}$ and every transition is labelled by a timed atomic action ($\neq \delta$). The operational semantics given in this section concerns two relations:

$$\begin{array}{lll} \mathit{Step} & \subseteq & \mathcal{T} \times A^{time} \times \mathcal{T} \\ \mathit{Terminate} & \subseteq & \mathcal{T} \times A^{time} \end{array}$$

These relations are defined as the least relations satisfying the action rules of Table 4. We write:

$$\begin{array}{lll} p \xrightarrow{a(r)} p' & \text{for} & (p, a(r), p') \in \mathit{Step} \\ p \xrightarrow{a(r)} \surd & \text{for} & (p, a(r)) \in \mathit{Terminate} \end{array}$$

Of course, it would be possible to define *State* as $\mathcal{T} \cup \{\surd\}$, in which case the relation *Terminate* would not be needed anymore. However, in this section we follow the convention of [BB89].

If we say that the term p has a certain transition, we mean that the transition system associated to p has this transition.

We have to adapt the definition of strong bisimulation, since $p \xrightarrow{a(r)} \surd$ is only an abbreviation of $(p, a(r)) \in \mathit{Terminate}$ and we need an extra condition, concerning the *ultimate delay* of two related terms. We formulate the following definition:

Definition 3.2.1 *ultimate delay bisimulation*
Let $p, q \in \mathcal{T}$, then $p \leftrightarrow_U q$ iff there is a relation $R \subseteq \mathcal{T} \times \mathcal{T}$ such that:

1. *the roots p and q are related by R*

2. **if** *$s \xrightarrow{a(r)} s'$ is a transition in the transition system of p and $R(s,t)$*
   then *there is a transition $t \xrightarrow{a(r)} t'$ in the transition system of q such that $R(s',t')$*

3. **if** *$t \xrightarrow{a(r)} t'$ is a transition in the transition system of q and $R(s,t)$*
   then *there is a transition $s \xrightarrow{a(r)} s'$ in the transition system of p such that $R(s',t')$*

$a \in A,\ r > 0$

$$atom \quad : \quad a(r) \xrightarrow{a(r)} \surd$$

$$\cdot \quad : \quad \frac{p \xrightarrow{a(r)} \surd}{p \cdot q \xrightarrow{a(r)} r \gg q} \qquad \frac{p \xrightarrow{a(r)} p'}{p \cdot q \xrightarrow{a(r)} p' \cdot q}$$

$$+ \quad : \quad \frac{p \xrightarrow{a(r)} \surd}{p + q \xrightarrow{a(r)} \surd,\quad q + p \xrightarrow{a(r)} \surd}$$

$$\frac{p \xrightarrow{a(r)} p'}{p + q \xrightarrow{a(r)} p',\quad q + p \xrightarrow{a(r)} p'}$$

$$\gg \quad : \quad \frac{s < r \quad p \xrightarrow{a(r)} \surd}{s \gg p \xrightarrow{a(r)} \surd} \qquad \frac{s < r \quad p \xrightarrow{a(r)} p'}{s \gg p \xrightarrow{a(r)} p'}$$

Table 4: Transition System Specification for BPA$\rho\delta$

4. **if** $s \xrightarrow{a(r)} \surd$ *is a terminating transition in the transition system of* p *and* $R(s,t)$ **then** *there is a terminating transition* $t \xrightarrow{a(r)} \surd$ *in the transition system of* q
5. **if** $t \xrightarrow{a(r)} \surd$ *is a terminating transition in the transition system of* q *and* $R(s,t)$ **then** *there is a terminating transition* $s \xrightarrow{a(r)} \surd$ *in the transition system of* q
6. **if** s *and* t *are related* **then** *they must have the same ultimate delay*

Lemma 3.2.2 $\forall p \in \mathcal{T}\ \forall r \in \mathbb{R}^{\geq 0}$

$$r \in inittim(p) \iff \exists a\, (\exists p' \quad p \xrightarrow{a(r)} p' \vee p \xrightarrow{a(r)} \surd)$$

Proof.
$\Longrightarrow$ By induction on the *size* of p;
$\Longleftarrow$ By induction on the length of the derivation. □

Lemma 3.2.3 $\forall p, q \in \mathcal{T}$

$$p \underline{\leftrightarrow}_U q \implies U(p) = U(q) \wedge L(p) = L(q)$$

Proof. Assume $p \underline{\leftrightarrow}_U q$, then $U(p) = U(q)$ follows by definition. $L(p) = L(q)$ follows from the previous Lemma and the definition of $\underline{\leftrightarrow}_U$ since $L(p) = max(inittim(p))$ □

An axiom is sound w.r.t. this operational semantics if two arbitrary terms which are instantiations of the axiom have bisimilar transition systems, according to the definition 3.2.1. In the next theorem the soundness of the axioms of BPA$\rho\delta$ is stated. Note that for the soundness of the theory BPA$\rho\delta$ itself, we need that bisimulation equivalence is a congruence as well. In the sequel we extend the theory BPA$\rho\delta$ resulting in new soundness theorems, but only in this case we provide a formal proof. Hence, the proof of the next theorem serves as as an example how to prove the soundness of a set of axioms w.r.t. to an operational semantics.

Theorem 3.2.4 *The axioms of* BPA$\rho\delta$ *are sound w.r.t. to the ultimate delay bisimulation equivalence* $\leftrightarrow_U$.

Proof. See Appendix A □

3.3 An Abstract Operational Semantics with δ-Rules

The operational semantics of [BB89] has the notion of an *idle* transition. In this operational semantics every state consists of an expression part and a time part; an *idle* transition increases the time part while the expression part remains the same. In this way a possible deadlock is reflected; the transition system of $a(2) + \delta(5)$ has uncountably many *idle* transitions before 5. In the operational semantics presented in the previous section we abstract from these *idle* transitions by adding an extra condition to the notion of bisimulation saying that bisimilar terms must have the same *ultimate delay* . In this way, terms are distinguished as well if they differ only in their deadlock behaviour.

Another possibility is to introduce a relation

$$Deadlock \subseteq \mathcal{T} \times \mathbb{R}^{\geq 0}$$

containing all pairs (p,r) such that p has a deadlock at time r. Hence, $\{(\delta(5),5),(6 \gg b(4),6)\} \subset$ *Deadlock*. We write

$$p \xrightarrow{\delta(r)} \surd \quad \text{for} \quad (p,r) \in Deadlock$$

or we say "p has a $\delta(r)$-transition". We say "p has a δ-transition" if we are not interested in the timestamp of the deadlock. Since terms differing only in their deadlock behaviour have different δ-transitions now, we do not need anymore the extra condition concerning the *ultimate delay* of related terms. The notions of *State*,*Label* and the definitions of *Step* and *Terminate* remain as in the previous section. Also the rules for $a \neq \delta$ remain the same as in the previous section, therefore we may "import" these rules. A δ-rule is an action rule which generates δ-transitions. A term has a possible deadlock whenever its *ultimate delay* $U(p)$ is later than the latest possible action, denoted by $L(p)$. The rules are given in Table 5.

Import Table 4
$\dfrac{U(p) > L(p)}{p \xrightarrow{\delta(U(p))} \surd}$

Table 5: Transition System Specification for BPA$\rho\delta$ with δ-rule

Definition 3.3.1 *strong bisimulation*
Let $p, q \in State$ then $p \leftrightarrow q$ iff there is a relation $R \subseteq State \times State$ such that:

1. *the roots* p *and* q *are related by* R
2. **if** $s \xrightarrow{a(r)} s'$ *is a transition in the transition system of* p *and* $R(s,t)$ **then** *there is a transition* $t \xrightarrow{a(r)} t'$ *in the transition system of* q *such that* $R(s',t')$
3. **if** $t \xrightarrow{a(r)} t'$ *is a transition in the transition system of* q *and* $R(s,t)$ **then** *there is a transition* $s \xrightarrow{a(r)} s'$ *in the transition system of* p *such that* $R(s',t')$
4. **if** $s \xrightarrow{a(r)} \surd$ *is a terminating transition in the transition system of* p *and* $R(s,t)$ **then** *there is a terminating transition* $t \xrightarrow{a(r)} \surd$ *in the transition system of* q
5. **if** $t \xrightarrow{a(r)} \surd$ *is a terminating transition in the transition system of* q *and* $R(s,t)$ **then** *there is a terminating transition* $s \xrightarrow{a(r)} \surd$ *in the transition system of* q
6. **if** $s \xrightarrow{\delta(r)} \surd$ *is a deadlock transition in the transition system of* p *and* $R(s,t)$ **then** *there is a deadlock transition* $t \xrightarrow{\delta(r)} \surd$ *in the transition system of* q
7. **if** $t \xrightarrow{\delta(r)} \surd$ *is a deadlock transition in the transition system of* q *and* $R(s,t)$ **then** *there is a deadlock transition* $s \xrightarrow{\delta(r)} \surd$ *in the transition system of* q

Note that this additional δ-rule induces the relation *Deadlock* $\subseteq \mathcal{T} \times \mathbb{R}^{\geq 0}$. Since we import the action rules of Table 4 as well, the relations *Step* and *Terminate* are imported as well. Next, we give some simple Lemma's.

Lemma 3.3.2 $\forall p \in \mathcal{T}\ \forall u \in \mathbb{R}^{\geq 0}$

$$p \xrightarrow{\delta(u)} \surd \iff U(p) = u \wedge U(p) > L(p)$$

Proof. Directly from the action rules for δ transitions. □

Lemma 3.3.3 $\forall p, q \in \mathcal{T}$

$$p \leftrightarroweq q \implies U(p) = U(q) \wedge L(p) = L(q)$$

Proof. Assume $p \leftrightarroweq q$. $L(p) = L(q)$ follows from the Lemma 3.2.2 and Definition 3.3.1. If $U(p) > L(p)$ then $p \xrightarrow{\delta(U(p))} \surd$ and by Definition 3.3.1 $q \xrightarrow{\delta(U(p))} \surd$ as well and we may conclude $U(p) = U(q) > L(q)$. If $U(p) = L(p)$ then $U(p) = U(q)$ follows from $L(p) = L(q)$ and $U(p) > L(p) \iff U(q) > L(q)$. □

Lemma 3.3.4 *Correspondence Lemma for basic terms w.r.t. the operational semantics of Table 5*

$\forall p \in \mathcal{B}\ \forall a \in A\ \forall r \in \mathbb{R}^{>0}$

$$\begin{array}{lcc} p \xrightarrow{a(r)} \surd & \iff & a(r) \sqsubseteq p \\ p \xrightarrow{a(r)} r \gg p' & \iff & a(r) \cdot p' \sqsubseteq p \\ p \xrightarrow{\delta(r)} \surd & \iff & \delta(r) \sqsubseteq p \end{array}$$

Proof.
$\Longrightarrow$ By induction on the length of a derivation of the transition.
$\Longleftarrow$ By the soundness of the axioms $A1$ and $A2$ and the definition of $\sqsubseteq$. □

For *basic* terms we have a Lemma which states that the *(absolute) time shift* operator does not filter anything away in the action rule for sequential composition, in other words, *basic* terms have ascending timestamps:

Lemma 3.3.5 *Basic terms have ascending timestamps*
$\forall p \in \mathcal{B}$

$$p \xrightarrow{a(r)} r \gg p' \quad \Longrightarrow \quad r \gg p' \leftrightarrow p'$$

Proof. From the action rules we know that if $q \xrightarrow{a(t)} q'$ with $t > s$ then $s \gg q \xrightarrow{a(t)} q'$ as well. By definition of *basic* terms and $a(r) \cdot p' \sqsubseteq p$ we know $p' \in \mathcal{B}\ (r)$. Hence, every transition of p' has a timestamp greater than r. And thus, if $p' \xrightarrow{b(t)} p$" then $r \gg p' \xrightarrow{b(t)} p$" as well for all b in A. Simarly for terminating and deadlocking transitions of p'. □

The consequence of this Lemma is, that for *basic* terms the application of the *(absolute) time shift* is not necessary anymore in the action rule for sequential composition, thus the action rule

$$\frac{p \xrightarrow{a(r)} \surd}{p \cdot q \xrightarrow{a(r)} r \gg q} \quad \text{can be simplified to} \quad \frac{p \xrightarrow{a(r)} \surd}{p \cdot q \xrightarrow{a(r)} q}$$

Hence, the action rules for the semantics of *ACP* ([Gla87]) are sufficient for *basic* terms.

Lemma 3.3.6 *The operational semantics of the previous section is equivalent with the one above, in the sense that for all $p, q \in \mathcal{T}$ and $R \subseteq \mathcal{T} \times \mathcal{T}$:*

$$\begin{array}{ll} p \leftrightarrow_U q & \textit{according to } R \\ & \Longleftrightarrow \\ p \leftrightarrow q & \textit{according to } R \end{array}$$

Proof.

- $\Longrightarrow$ By induction on the *size* of p, so assume it has been proven for smaller states and consider $(p, q) \in R$. Then there are two cases
  - $U(p) = L(p)$ then $U(p) = U(q) = L(q) = L(p)$ and neither p nor q has a deadlock. Since both definitions are equivalent for terms without deadlocks we may conclude $p \leftrightarrow q$.
  - $U(p) > L(p)$, but then also $U(p) = U(q) > L(q) = L(p)$ and we have $(u = U(p))$

    $$p \xrightarrow{\delta(u)} \surd \quad \text{and} \quad q \xrightarrow{\delta(u)} \surd$$

    Since both definitions are equivalent with respect to non-deadlock transitions and p and q have the same deadlock transition we may conclude $p \leftrightarrow q$.
- $\Longleftarrow$ Analogously

□

Equivalence with the Original Operational Semantics

In this section we show that the abstract operational semantics without time component as presented in this report is equivalent with the original semantics from [BB89] as given in section 1. Remember that in the original semantics every state has both an expression component and a time-component. Consider

$$\begin{array}{lccc} \text{In abstract operational semantics} & (a(r)\cdot p)\cdot q & \xrightarrow{a(r)} & (r \gg p)\cdot q \\ \text{In original operational semantics} & <(a(r)\cdot p)\cdot q, 0> & \xrightarrow{a(r)} & <p\cdot q, r> \end{array}$$

We see that in the abstract semantics the course of time is encoded by an occurrence of $r \gg$ somewhere in the prefix, while in the original semantics the course of time is being reflected by the time component in the state.

In order to relate the states of the (abstract) transition system of p with the states of the (original) transition $<p, 0>$, we define two functions inductively.

$$\begin{array}{lcllcl} strip(a(r)) & = & a(r) & time(a(r)) & = & 0 \\ strip(X+Y) & = & X+Y & time(X+Y) & = & 0 \\ strip(X\cdot Y) & = & strip(X)\cdot Y & time(X\cdot Y) & = & time(X) \\ strip(r \gg X) & = & X & time(r \gg X) & = & r \end{array}$$

Now we can state some simple Lemma's concerning the (abstract) transition system $\longrightarrow \subseteq \mathcal{T}\times\mathcal{T}$:

Every resulting state contains an occurrence of $r \gg$ in its prefix, which can be shown by the following Lemma:

Lemma 3.3.7 $\forall p, p' \in \mathcal{T}\ \forall a \in A\ \forall r \in \mathbb{R}^{>0}$

$$p \xrightarrow{a(r)} p' \quad \Longrightarrow \quad time(p') = r$$

Proof. By induction on the length of the derivation. □

Moreover, we have

Lemma 3.3.8 $\forall p, p' \in \mathcal{T}\ \forall a \in A\ \forall r \in \mathbb{R}^{>0}$

$$p \xrightarrow{a(r)} p' \quad \Longleftrightarrow \quad r > time(p) \ \wedge\ strip(p) \xrightarrow{a(r)} p'$$

Proof. Both directions can be proven by induction on the length of the derivation. □

The following Lemma states a correspondence between the transition system of p in the abstract operational semantics and the transition system of $<strip(p), time(p)>$ in the original semantics:

Lemma 3.3.9 $\forall p, q \in \mathcal{T}\ \forall a \in A\ \forall r \in \mathbb{R}^{>0}$

$$\begin{array}{lcl} p \xrightarrow{a(r)} \surd & \Longleftrightarrow & <strip(p), time(p)> \xrightarrow{a(r)} <\surd, r> \\ p \xrightarrow{a(r)} q & \Longrightarrow & <strip(p), time(p)> \xrightarrow{a(r)} <strip(q), r> \\ <strip(p), time(p)> \xrightarrow{a(r)} <q, r> & \Longrightarrow & \exists q' \quad strip(q') \equiv q \ \wedge\ time(q') = r \ \wedge\ p \xrightarrow{a(r)} q' \end{array}$$

Proof. All cases can be proven by induction on the length of the derivation. In some cases the previous Lemma must be used. □

Furthermore, w.r.t. the original semantics we know that $U(p) = sup\{s | <p, t> \longrightarrow <p, s>\}$ and that $U(p) > L(p)$ implies that $<p, t> \stackrel{b(U(p))}{\not\longrightarrow}$ for all b in A.

Now we are ready to prove the following:

Theorem 3.3.10 *The operational semantics presented in this section for* BPA$\rho\delta$ *is equivalent with the original semantics for* ACPρI*:*
$\forall p, q \in \mathcal{T}$

$$p \leftrightarrow q \Longleftrightarrow < p, 0 > \; \underline{\leftrightarrow} \; < q, 0 >$$

Proof. First we prove by induction on $(size(p), size(q))$ that

$$\forall p, q \in \mathcal{T} \quad time(p) = time(q) : \quad p \leftrightarrow q \quad \Longleftrightarrow \quad < strip(p), time(p) > \; \underline{\leftrightarrow} \; < strip(q), time(q) > .$$

Next, we have to show both directions of $p \leftrightarrow q \Longleftrightarrow < p, 0 > \; \underline{\leftrightarrow} \; < q, 0 >$:

- $\Longrightarrow$ Assume $R \subseteq \mathcal{T} \times \mathcal{T}$ is a bisimulation relation containing (p, q). Note that for all $p_0, q_0 \in \mathcal{T}$ $(p_0, q_0) \in R$ implies $U(p_0) = U(q_0)$. Construct $R^* \subseteq (\mathcal{T} \times \mathbb{R}^{\geq 0}) \times (\mathcal{T} \times \mathbb{R}^{\geq 0})$:

$$\begin{array}{rcl} R^* & = & \{ \; (< strip(p_1), r >, < strip(q_1), r >) \mid (p_1, q_1) \in R \; \wedge \; 0 \leq r < U(p_1) \; \} \\ & \cup & \{ \; (< p, r >, < q, r >) \mid 0 \leq r < U(p) \; \} \end{array}$$

  The proof that R^* is a bisimulation relation is left to the reader.

- $\Longleftarrow$ Assume $R^* \subseteq (\mathcal{T} \times \mathbb{R}^{\geq 0}) \times (\mathcal{T} \times \mathbb{R}^{\geq 0})$ is a bisimulation relation containing $(< p, 0 >, < q, 0 >)$, then construct $R \subseteq \mathcal{T} \times \mathcal{T}$:

$$\begin{array}{rcl} R & = & \{ \; (p_0, q_0) \mid (< strip(p_0), time(q_0) >, < strip(q_0), time(q_0) >) \in R^* \; \} \\ & \cup & \{ \; (p, q) \; \} \end{array}$$

  Again, the proof that R is a bisimulation relation is left to the reader.

□

3.4 An Abstract Operational Semantics for BPA$\rho\delta$ in *ntyft* format

In the previous section we introduced a model for BPA$\rho\delta$ by giving a Transition System Specification and we stated the soundness of the axioms of BPA$\rho\delta$. In order to obtain the soundness of the theory BPA$\rho\delta$ we need to prove as well that bisimulation equivalence is a congruence. We do this by using a result of [Gro89]. Groote proved in this paper that if the action rules of a Transition System Specification are in a specific format, which he called the *ntyft/ntyxt*-format, then bisimulation equivalence is a congruence. In appendix B a short introduction in the ideas and results of [Gro89] is given. In this section we give a Transition System Specification in *ntyft*-format, which is a instantiation of the *ntyft/ntyxt*-format. Since we prove as well that the induced bisimulation equivalence is equal to the one of the previous section we obtain the required congruence result.

A transition system specified by a Transition System Specification in *ntyft* format concerns only one relation. Therefore we cannot have a separate relation containing all the terminating transitions, instead we take the solution to encode termination in the label. (This idea can be found in [Bri88] and [BV89] as well)

We denote the transition relation, which is defined in this section, by $\longmapsto$ to distinguish it from the transition relation $\longrightarrow$ of the previous sections. We extend the set of labels with encodings of terminating transitions:

$$\begin{array}{rcl} A_{\surd} & = & \{a_{\surd} \mid a \in A\} \\ A_{\surd}^{time} & = & \{a_{\surd}(t) \mid a \in A \; \wedge \; t \in \mathbb{R}^{\geq 0}\} \\ \longmapsto & = & \mathcal{T} \times (A_{\delta}^{time} \cup A_{\surd}^{time}) \times \mathcal{T} \end{array}$$

Now we can encode in the label whether a transition is terminating or not. Instead of

$$p \xrightarrow{a(r)} \surd \text{ which is an abbreviation of } (p, a(r)) \in Terminate$$

we now have

$$p \xrightarrow{a_{\surd}(r)} \delta \text{ which is an abbreviation of } (p, a_{\surd}(r), \delta) \in \longmapsto$$

The next table, Table 6, contains a Transition System Specification for BPA$\rho\delta$ in *ntyft* format.

$a \in A,\ b \in A \cup A_{\surd},\ b' \in A_\delta \cup A_{\surd},\ r \in \mathbb{R}^{>0}$

atom : $a(r) \xmapsto{a_{\surd}(r)} \delta$

$\cdot$: $$\frac{p \xmapsto{a_{\surd}(r)} p'}{p \cdot q \xmapsto{a(r)} r \gg q} \qquad \frac{p \xmapsto{a(r)} p'}{p \cdot q \xmapsto{a(r)} p' \cdot q}$$

$+$: $$\frac{p \xmapsto{b(r)} p'}{p + q \xmapsto{b(r)} p', \quad q + p \xmapsto{b(r)} p'}$$

$\gg$: $$\frac{s < r \quad p \xmapsto{b'(r)} p'}{s \gg p \xmapsto{b'(r)} p'}$$

$r \in \mathbb{R}^{>0}$

atom : $\delta(r) \xmapsto{\delta(r)} \delta$

$\cdot$: $$\frac{p \xmapsto{\delta(r)} p'}{p \cdot q \xmapsto{\delta(r)} p'} \qquad \frac{\forall b \in A_\delta \cup A_{\surd}\ \forall r > s\ p \not\xmapsto{b(r)}}{s \gg p \xmapsto{\delta(s)} \delta}$$

$+$: $$\frac{p \xmapsto{\delta(r)} p' \quad \forall b \in A \cup A_{\surd}\ \forall r' \geq r\ q \not\xmapsto{b(r')} \quad \forall s > r\ q \not\xmapsto{\delta(s)}}{p + q \xmapsto{\delta(r)} p', \quad q + p \xmapsto{\delta(r)} p'}$$

Table 6: Transition System Specification for BPA$\rho\delta$ in *ntyft* format

Definition 3.4.1 *strong bisimulation for* $\longmapsto$
Let $p, q \in T$ *then* $p \underline{\leftrightarrow}_{\longmapsto} q$ *iff there is a relation* $R \subseteq T \times T$ *such that (a is taken from* $A_\delta \cup A_{\surd}$*):*

1. *the roots p and q are related by R*
2. if $s \xrightarrow{a(r)} s'$ *is a transition in the transition system of p and* $R(s,t)$
   then *there is a transition* $t \xrightarrow{a(r)} t'$ *in the transition system of q such that* $R(s',t')$
3. if $t \xrightarrow{a(r)} t'$ *is a transition in the transition system of q and* $R(s,t)$
   then *there is a transition* $s \xrightarrow{a(r)} s'$ *in the transition system of p such that* $R(s',t')$

Theorem 3.4.2 *Bisimulation Equivalence is a congruence*

Proof. We use a result of [Gro89]; if the Transition System Specification is well founded and the action rules are in a specific format, called the *ntyft* format, and the action rules can be stratified by a stratification function, then bisimulation equivalence is a congruence. A short treatment of this is given in Appendix B, however for a full understanding we refer to [Gro89]. Since the action rules of Table 6 are in the *ntyft* format it is sufficient to give a stratification. Instead of saying that we have one operator $\gg$ which takes a real number and a process term we can say also that for each $r \in \mathbb{R}^{\geq 0}$ there is an operator $r \gg: \mathcal{T} \longrightarrow \mathcal{T}$. Moreover δ can be seen as a constructor with rank zero. A stratification that does the job is the following:

$$\begin{array}{ll} \forall p, p', r, a \neq \delta & n(p \xmapsto{a(r)} p') = 0 \\ \forall p, p', r & n(p \xmapsto{\delta(r)} p') = size(p) \end{array}$$

□

Lemma 3.4.3 $\forall p \in \mathcal{T}\ \forall u \in \mathbb{R}$

$$u = U(p) > L(p) \iff p \xmapsto{\delta(u)} \delta$$

Proof. See Appendix C □

The equivalence of the transition systems $\longrightarrow$ and $\longmapsto$

First we state the following Lemma

Lemma 3.4.4 $\forall p, p' \in \mathcal{T}\ \forall a \in A\ \forall r \in \mathbb{R}^{>0}$

$$\begin{array}{lll} p \xrightarrow{a(r)} p' & \iff & p \xmapsto{a(r)} p' \\ p \xrightarrow{a(r)} \surd & \iff & p \xmapsto{a\surd(r)} \delta \\ p \xrightarrow{\delta(r)} \surd & \iff & p \xmapsto{\delta(r)} \delta \end{array}$$

Proof. Left to the reader □

From this Lemma we conclude

Theorem 3.4.5 $\forall p, q \in \mathcal{T}$

$$p \underline{\leftrightarrow} q \iff p \underline{\leftrightarrow}_{\longmapsto} q$$

Proof. From Lemma 3.4.4 it follows $R \subset \mathcal{T} \times \mathcal{T}$ containing (p, q) is a bisimulation relation according to Definition 3.3.1 (for $\underline{\leftrightarrow}$) iff it is a bisimulation relation according to Definition 3.4.1 (for $\underline{\leftrightarrow}_{\longmapsto}$).
□

3.5 Completeness of BPA$\rho\delta$

Since we know that all the semantics which have been given so far, are equivalent with each other, it does not matter for which one we prove the completeness. We will continue with the semantics of the bisimulation equivalence $\underline{\leftrightarrow}$ associated with the Transition System Specification with δ-rules of Table 5.

The theory BPA$\rho\delta$ is complete w.r.t. the operational semantics if two arbitrary terms have bisimilar transition systems.

Theorem 3.5.1 *Completeness*
$\forall p, q \in \mathcal{T}$

$$p \underline{\leftrightarrow} q \Longrightarrow \text{BPA}\rho\delta \vdash p = q$$

It is sufficient to consider *basic* terms only.
Proof. First we will prove by induction on the *size* of p that $p \subseteq q$. Assume

$$p \simeq \textstyle\sum_i a_i(r_i) \cdot p_i + \sum_j b_j(s_j)$$
$$q \simeq \textstyle\sum_k c_k(t_k) \cdot q_k + \sum_l d_l(u_l)$$

$\forall i$	$p \xrightarrow{a_i(r_i)} r_i \gg p_i$	by bisimulation also q has a transition
$\exists q'$	$q \xrightarrow{a_i(r_i)} r_i \gg q'$	with $r_i \gg p_i \underline{\leftrightarrow} r_i \gg q'$.
		Moreover, p and q are *basic* terms, hence
	$p_i \underline{\leftrightarrow} r_i \gg p_i \underline{\leftrightarrow} r_i \gg q' \underline{\leftrightarrow} q'$	and by transitivity of $\underline{\leftrightarrow}$ and by induction
	$p_i = q'$	together with
	$a_i(t_i) \cdot q' \sqsubseteq q$	we may conclude
	$a_i(t_i) \cdot p_i \subseteq q$	
$\forall j$	$p \xrightarrow{b_j(s_j)} \surd$	by bisimulation also q has a transition
	$q \xrightarrow{b_j(s_j)} \surd$	and thus
	$b_j(s_j) \sqsubseteq q$	

Adding these results together we get $p \subseteq q$. Since bisimulation is symmetric we may conclude that $q \subseteq p$ as well. From $p \subseteq q \ \wedge \ q \subseteq p$ we conclude BPA$\rho\delta \ \vdash p = q$. □

4 Parallelism and Synchronisation

4.1 Introduction

Now we are ready to introduce parallelism and synchronisation resulting in the theory ACPρ from [BB89]. 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 our real time setting it is a bit more subtle; if we take for example the process $(a(t) \cdot X) \mathbin{\|\!\_} Y$ then by executing the a action on time t the whole process must proceed in time. Whenever Y can wait after t (if $t < U(Y)$) then $(a(t) \cdot X) \mathbin{\|\!\_} Y$

can execute a action on time t, resulting in $(t \gg X)\|(t \gg Y)$, otherwise (if $t \geq U(Y)$) a deadlock at time $U(Y)$ occurs. Consider:

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

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 at the *ultimate delay* of the right component is the result. The point is that in the process denoted by $X \mathbin{⌊\!⌊} Y$ only those initial actions of X can be executed, which have a timestamp smaller than $U(Y)$. Hence, we need an auxiliary operator, called the *bounded initialization* . The *bounded initialization* has as input a process X and a time t and delivers that part of X which starts before t. This new operator is the counterpart of the *(absolute) time shift* operator, which is denoted $t \gg X$, as presented earlier. Therefore we shall denote the *bounded initialization* of X to t by $X \gg t$.

Definition 4.1.1 *$\mathcal{T}$, with typical elements p, p_1, p_2, is the set of terms over* ACPρ, *where $a \in A_\delta$, $r \in \mathbb{R}^{\geq 0}$ and $H \subseteq A$:*

$$p \in \mathcal{T} \quad p := a(r) \mid p_1 + p_2 \mid p_1 \cdot p_2 \mid r \gg p \mid p \gg r \mid p_1 \| p_2 \mid p_1 \mathbin{⌊\!⌊} p_2 \mid p_1 | p_2 \mid \delta_H(p)$$

We have to extend the definition of the syntactic operators for terms with parallelism, left merge and synchronisation. *initact*(p) is an auxiliary operator, which delivers the set of initial timed actions which can be executed by p. We need this operator since the the timestamps of the actions which can be executed by $p|q$ are dependent on the timed actions of p and q and not only on the associated timestamps.

Definition 4.1.2 *Definition of U(p), initact(p), inittim(p), L(p). For U(p) only the additional rules are given.*

$a \in A$

$$\begin{aligned}
U(X \gg r) &= min(U(X), r) \\
U(X \| Y) &= min(U(X), U(Y)) \\
U(X \mathbin{⌊\!⌊} Y) &= min(U(X), U(Y)) \\
U(X | Y) &= min(U(X), U(Y)) \\
U(\delta_H(X)) &= U(X)
\end{aligned}$$

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

$$inittim(X) = \{t | \exists a \in A \ : \ a(t) \in initact(p)\}$$

$$L(X) = max(inittim(X)) \text{ where } max(\emptyset) = 0$$

We do not extend the definitions of $\sqsubseteq$ and $\subseteq$, hence these relations are defined for terms over BPA$\rho\delta$ only. We don't need these relations in the sequel.

We assume a communication function $| : A_\delta \times A_\delta \longrightarrow A_\delta$ which is symmetric, commutative and associative and has δ as zero element. We don't need time information in the definition of this communication function because in our view it does not make sense to define a communication between atomic actions which happen at a different time. Thus if $\bar{a}|a = c$ then

$$\begin{array}{rcl} \bar{a}(2)|a(2) & = & c(2) \\ \bar{a}(1)|a(3) & = & \delta(1) \end{array}$$

4.2 The Theory ACPρ

The axiom system for ACPρ is given in the Table 7, some remarks on these axioms are given below.

$a, b \in A_\delta$		
ATB5	$t \geq r$	$a(t) \gg r = \delta(r)$
ATB6	$t < r$	$a(t) \gg r = a(t)$
ATB7		$(X + Y) \gg t = (X \gg t) + (Y \gg t)$
ATB8		$(X \cdot Y) \gg t = (X \gg t) \cdot Y$
ATC1	$t \neq r$	$a(t)\vert b(r) = \delta(min(t, r))$
ATC2		$a(t)\vert b(t) = (a\vert b)(t)$
CM1		$X\Vert Y = X \mathbin{\|\!\_} Y + Y \mathbin{\|\!\_} X + X\vert Y$
ATCM2		$a(t) \mathbin{\|\!\_} Y = (a(t) \gg U(Y)) \cdot Y$
ATCM3		$(a(t) \cdot X) \mathbin{\|\!\_} Y = (a(t) \gg U(Y)) \cdot (X\Vert Y)$
CM4		$(X_1 + X_2) \mathbin{\|\!\_} Y = X_1 \mathbin{\|\!\_} Y + X_2 \mathbin{\|\!\_} Y$
CM5		$(a(t) \cdot X)\vert b(r) = (a(t)\vert b(r)) \cdot X$
CM6		$a(t)\vert(b(r) \cdot Y) = (a(t)\vert b(r)) \cdot Y$
CM7		$(a(t) \cdot X)\vert(b(r) \cdot Y) = (a(t)\vert b(r)) \cdot (X\Vert Y)$
CM8		$(X_1 + X_2)\vert Y = X_1\vert Y + X_2\vert Y$
CM9		$X\vert(Y_1 + Y_2) = X\vert Y_1 + X\vert Y_2$
D1	$a \notin H$	$\delta_H(a) = a$
D2	$a \in H$	$\delta_H(a) = \delta$
ATD		$\delta_H(a(t)) = (\delta_H(a))(t)$
D3		$\delta_H(X + Y) = \delta_H(X) + \delta_H(Y)$
D4		$\delta_H(X \cdot Y) = \delta_H(X) \cdot \delta_H(Y)$

Table 7: ACPρ =BPA$\rho\delta$ +ATB5-8+ATC1,2+CM1+ATCM2,3+CM4-9+D1-4+ATD

ATC1 Both processes have to wait in parallel. Synchronisation is not possible when the time stamps are not equal and a deadlock occurs at the moment that one of the initial actions is not active any longer.

ATC2 If the time stamps are equal then synchronisation may occur according to the communication function.

CM1 This axiom is exactly the same as in ACP, however together with the axioms for the left merge it does not result in arbitrary interleaving. Now the time stamps of the atomic actions determine the possible orderings.

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

ATCM2,3 In a left merge we have to ensure that the right component can wait long enough, otherwise the left merge delivers a deadlock a the moment that the right component cannot wait any longer.

$$\begin{aligned} a(2) \,⌊\!⌊\, b(3) &= (a(2) \gg 3)\cdot b(3) \\ &= a(2)\cdot b(3) \\ b(3) \,⌊\!⌊\, a(2) &= (b(3) \gg 2)\cdot a(2) \\ &= \delta(2)\cdot a(2) \\ &= \delta(2) \end{aligned}$$

Note that the definition of the *ultimate delay* operator is extended to all operators, which is not the case in [BB89]. Moreover, we can check that the definitions of all the syntactic operators are consistent with the theory, which means for example that ACPρI $\vdash X = Y \Longrightarrow U(X) = U(Y)$

Lemma 4.2.1 *For each $p, q \in \mathcal{B}$ there is a $z \in \mathcal{B}$ such that* ACP$\rho \vdash z = p\|q$

Proof. Assume

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

We proof the Lemma by induction on the *size* of p. We give only the atomic case, the other case can be done analogously. Define the following subsets

$$\begin{aligned} J^t &= \{j \in J \mid s_j > t\} \\ L^t &= \{l \in L \mid u_l > t\} \\ J_L &= \{j \in J \mid L^{s_j} \neq \emptyset\} \\ L_J &= \{l \in L \mid J^{u_l} \neq \emptyset\} \end{aligned}$$

We can construct a $z \in \mathcal{B}$ such that $z = p\|q$ by taking

$$z \simeq \sum_{j\in J_L} b_j(s_j)\cdot \sum_{l\in L^{s_j}} d_l(u_l) \;+\; \sum_{l\in L_J} d_l(u_l)\cdot \sum_{j\in J^{u_l}} b_j(s_j) \;+\; \sum_{j,l\in J\times L: s_j=u_l} (b_j|d_l)(s_j)$$

□

Lemma 4.2.2 *For each $p \in \mathcal{B}$ there is a $z \in B$ such that* ACP$\rho \vdash z = \delta_H(p)$

Proof. By simple induction on the *size* of p. □

Theorem 4.2.3 *Elimination Theorem*
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 have to discuss four cases, the other cases can be proved as in Lemma 2.2.4:

1. $p \equiv q|r$ By induction there are $q_b, r_b \in \mathcal{B}$ such that ACP$\rho \vdash q = q_b$ and ACP$\rho \vdash r = r_b$ with

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

After applying CM8 and CM9 sufficiently many times we get

$$\begin{array}{rclclc} p & = & \sum_{(i,k)\in I\times K} a_i(r_i)\cdot q_i|c_k(t_k)\cdot r_k & + & \sum_{(i,l)\in I\times L} a_i(r_i)\cdot q_i|d_l(u_l) & + \\ & & \sum_{(j,k)\in J\times K} b_j(s_j)|c_k(t_k)\cdot r_k & + & \sum_{(j,l)\in J\times L} b_j(s_j)|d_l(u_l) & \\ & \overset{\text{CM5-7}}{=} & \sum_{(i,k)\in I\times K}(a_i(r_i)|c_k(t_k))\cdot(q_i\|r_k) & + & \sum_{(i,l)\in I\times L}(a_i(r_i)|d_l(u_l))\cdot q_i & + \\ & & \sum_{(j,k)\in J\times K}(b_j(s_j)|c_k(t_k))\cdot r_k & + & \sum_{(j,l)\in J\times L} b_j(s_j)|d_l(u_l) & \end{array}$$

Each $a(t)|b(t')$ reduces to some $c(t)$ or $\delta(min(t,t'))$ and we know that a term not containing $\|$, $⫫$ or $|$ has a *basic* term. By the previous Lemma 4.2.1 there is for each $(i,k) \in I\times K$ a $p_{(i,j)} \in \mathcal{B}$ such that ACP$\rho \vdash p_{(i,j)} = q_i\|r_k$. Construct the following subsets of index pairs:

$$\begin{array}{rcllll} \overline{I\times K} & = & \{ & (i,k)\in I\times K & | & a_i|c_k \neq \delta \wedge r_i = t_k \quad \} \\ \overline{I\times L} & = & \{ & (i,l)\in I\times L & | & a_i|d_l \neq \delta \wedge r_i = u_l \quad \} \\ \overline{J\times K} & = & \{ & (j,k)\in I\times K & | & b_j|c_k \neq \delta \wedge s_j = t_k \quad \} \\ \overline{J\times L} & = & \{ & (j,l)\in J\times L & | & b_j|d_l \neq \delta \wedge s_j = u_l \quad \} \end{array}$$

Next, construct $\overline{p}$:

$$\begin{array}{rclclc} \overline{p} & \simeq & \sum_{(i,k)\in\overline{I\times K}}(a_i|c_k)(r_i)\cdot p_{(i,j)} & + & \sum_{(i,l)\in\overline{I\times L}}(a_i|d_l)(r_i)\cdot q_i & + \\ & & \sum_{(j,k)\in\overline{J\times K}}(b_j|c_k)(s_j)\cdot r_k & + & \sum_{(j,l)\in\overline{J\times L}}(b_j|d_l)(s_j) & \end{array}$$

And we can conclude

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

2. $p \equiv q ⫫ r$ By induction there are $q_b, r_b \in \mathcal{B}$ such that ACP$\rho \vdash q = q_b$ and ACP$\rho \vdash r = r_b$ with

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

Define for arbitrary $t \in \mathbb{R}^{\geq 0}$

$$I^t = \{i | i \in I \wedge r_i < t\} \text{ and } J^t = \{j | j \in J \wedge s_j < t\}$$

After applying CM4 and ATCM2,3 sufficiently many times we get

$$p = \sum_{i\in I}(a_i(r_i) \gg U(r_b))\cdot(q_i\|r_b) + \sum_{j\in J}(b_j(s_j) \gg U(r_b))\cdot r_b$$

Take $U(r_b) = u$. By the previous Lemma 4.2.1 there is for each $i \in I^u$ a $p_i \in \mathcal{B}$ such that $p_i = q_i\|r_b$ and we can construct $\overline{p}$ (note $\forall i \in I^u : a_i(r_i) \gg u = a_i(r_i)$)

$$\overline{p} \simeq \sum_{i\in I^u} a_i(r_i)\cdot p_i + \sum_{j\in J^u} b_j(s_j)\cdot r_b$$

And we can conclude

$$p = \begin{cases} \delta(u) \in \mathcal{B} & \text{if } I^u \cup J^u = \emptyset \\ \overline{p} \in \mathcal{B} & \text{if } I^u \cup J^u \neq \emptyset \wedge (I - I^u)\cup(J - J^u) = \emptyset \\ \overline{p} + \delta(u) \in \mathcal{B} & \text{if } I^u \cup J^u \neq \emptyset \wedge (I - I^u)\cup(J - J^u) \neq \emptyset \end{cases}$$

3. $p \equiv q \| r$ This follows from CM1 and the first two cases

4. $p \equiv \delta_H(q)$ By induction there is a $q_b \in \mathcal{B}$ such that ACP$\rho \vdash q = q_b$, say

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

And the case follows by the previous Lemma 4.2.2.

□

4.3 An Abstract Operational Semantics for ACPρ with δ transitions

We add some rules to the operational semantics of BPA$\rho\delta$ for the new operators $\|$, $|$ and $\lfloor\!\lfloor$. Remember that if the process $a(r) \cdot p$ executes the $a(r)$ action, it evolves to $r \gg p$ denoting that the time has increased to r and that only the part of p starting after r may continue.

If $a(r) \cdot p$ executes this $a(r)$ in a parallel composition with q, the "increase its time" must hold for q as well. In other words, $(a(r) \cdot p) \| q$ evolves to $(r \gg p) \| (r \gg q)$ by executing a at time r whenever q can wait until r. The "check" whether q can increase its time, is denoted by $r < U(q)$. Hence, we obtain the following action rule:

$$\frac{p \xrightarrow{a(r)} p' \quad r < U(q)}{p \| q \xrightarrow{a(r)} p' \| (r \gg q)\ , \quad q \| p \xrightarrow{a(r)} (r \gg q) \| p'}$$

Note that if $p \xrightarrow{a(r)} p'$ then p' contains one or more occurrences of $r \gg$ (e.g. $time(p') = r$); hence, only transitions with a timestamp greater than r are allowed for p'. In the sequel we will extend the definitions of $strip()$ and $time()$ to terms with parallelism.

If there is a synchronisation then both components have to execute an action:

$$\frac{p \xrightarrow{a(r)} p' \quad q \xrightarrow{b(r)} q' \quad a|b = c \neq \delta}{p \| q \xrightarrow{c(r)} p' \| q'}$$

The rules for the *bounded initialization* are symmetric to the action rules for the *(absolute) time shift* as given in Table 4. The Transition System Specification of the operational semantics of ACPρ is given in Table 8. Bisimulation equivalence defined in Definition 3.3.1 is denoted by $\underline{\leftrightarrow}$.

Theorem 4.3.1 *The axioms of* ACPρ *are sound with respect to bisimulation equivalence.*

Proof. Straightforward □

Equivalence with the Original Operational Semantics

We extend the definition *strip* and *time* (see section 3.3):

$$\begin{array}{lll lll}
strip(X \| Y) & = & strip(X) \| strip(Y) & time(X \| Y) & = & \begin{cases} time(X) & \text{if } time(X) = time(Y) \\ undefined & \text{otherwise} \end{cases} \\
strip(X \lfloor\!\lfloor Y) & = & X \lfloor\!\lfloor Y & time(X \lfloor\!\lfloor Y) & = & 0 \\
strip(X | Y) & = & X | Y & time(X|Y) & = & 0 \\
strip(r \gg X) & = & X & time(r \gg X) & = & r
\end{array}$$

Now we need a Lemma which states that $time()$ is always defined in a resulting state:

Import rules from Table 4

$a \in A$

$\|, \lfloor\!\lfloor$:

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

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

$\|, |$:

$$\frac{p \xrightarrow{a(r)} p' \quad q \xrightarrow{b(r)} q' \quad a|b = c \neq \delta}{p\|q \xrightarrow{c(r)} p'\|q'\,, \quad p|q \xrightarrow{c(r)} p'\|q'} \qquad \frac{p \xrightarrow{a(r)} \surd \quad q \xrightarrow{b(r)} \surd \quad a|b = c \neq \delta}{p\|q \xrightarrow{c(r)} \surd\,, \quad p|q \xrightarrow{c(r)} \surd}$$

$$\frac{p \xrightarrow{a(r)} \surd \quad q \xrightarrow{b(r)} q' \quad a|b = c \neq \delta}{p\|q \xrightarrow{c(r)} q'\,, \quad q\|p \xrightarrow{c(r)} q'\,, \quad p|q \xrightarrow{c(r)} q'\,, \quad q|p \xrightarrow{c(r)} q'}$$

$\gg$:

$$\frac{s > r \quad p \xrightarrow{a(r)} p'}{p \gg s \xrightarrow{a(r)} p'} \qquad \frac{s > r \quad p \xrightarrow{a(r)} \surd}{p \gg s \xrightarrow{a(r)} \surd}$$

δ_H :

$$\frac{p \xrightarrow{a(r)} p' \quad a \notin H}{\delta_H(p) \xrightarrow{a(r)} \delta_H(p')} \qquad \frac{p \xrightarrow{a(r)} \surd \quad a \notin H}{\delta_H(p) \xrightarrow{a(r)} \surd}$$

$$\frac{U(p) > L(p)}{p \xrightarrow{\delta(U(p))} \surd}$$

Table 8: Transition System Specification for ACPρ

Lemma 4.3.2 $\forall p, p_0, p_1 \in \mathcal{T}\ \forall a \in A\ \forall r \in \mathbb{R}^{>0}$

$$p \xrightarrow{a(r)} p_0 \| p_1 \quad \Longrightarrow \quad time(p_0) = time(p_1) = r$$

Proof. We know already that if p is a term over BPA$\rho\delta$ then:

$$p \xrightarrow{a(r)} p' \quad \Longrightarrow \quad time(p') = r$$

and the Lemma follows direct by induction on the length of the derivation of $p \xrightarrow{a(r)} q_0 \| q_1$. □

Since the Lemma's 3.3.7, 3.3.8 and 3.3.9 hold for terms over ACPρ as well, we may conclude that the equivalence between $\longrightarrow$ and $\longmapsto$, as stated in Theorem 3.3.10, does still hold.

4.4 An Abstract Operational Semantics for ACPρ in *ntyft* format

In this section we follow the idea of section 3.4 by changing in every premise of the action rules of Table 8 all occurrences of $p \xrightarrow{a(r)} \surd$ into $p \xmapsto{a_{\surd}(r)} p'$. The action rules for transitions labelled with timed actions from A^{time} are given in Table 9. A more difficult task is to give δ-rules in *ntyft* format. These action rules are given in Table 10. In this Table we use a unary function $\hat{}$ on actions. If $b \in A_\delta$ then $\hat{b} = b$ and if $b = a_{\surd}$ for some $a \in A$ then $\hat{b} = a$. At first sight the action rules of Table 10 look rather complicated. But note that

$$\frac{U(p\|q) > L(p\|q)}{p\|q \xmapsto{\delta(U(p\|q))} \delta}$$

corresponds with the two rules

$$\frac{\{\ u = U(p) \quad u = U(q) \quad u \notin inittim(p\|q)\ \}}{p\|q \xmapsto{\delta(u)} \delta}\ , \qquad \frac{\{\ u = U(p) > L(p) \quad u \leq U(q)\ \}}{p\|q \xmapsto{\delta(u)} \delta\ , \quad q\|p \xmapsto{\delta(u)} \delta}$$

And these last two rules correspond exactly with the δ rules for $\|$ in Table 10.

Theorem 4.4.1 *Bisimulation Equivalence is a congruence*

Proof. By the Theorem 3.4.2, since the stratification which is given in the proof of Theorem 3.4.2 is still valid. □

Lemma 4.4.2 $\forall p \in \mathcal{T}\ \forall u \in \mathbb{R}^{\geq 0}$

$$u = U(p) > L(p) \quad \Longleftrightarrow \quad p \xmapsto{\delta(u)} \delta$$

Proof. See Appendix C. □

Theorem 4.4.3 *The two Transition System Specifications and the associated definitions of bisimulation induce the same equivalence.*
$\forall p, q \in \mathcal{T}$

$$p \leftrightarroweq q \Longleftrightarrow p \leftrightarroweq_{\longmapsto} q$$

Proof. Follows directly from the fact that 3.4.4 holds for terms over ACPρ as well. □

Import rules from Table 6

$a, a' \in A,\ b \in A_\delta \cup A_{\surd}$

$\|, \mathbin{\|\!\lfloor}$:

$$\frac{p \xmapsto{a(r)} p' \quad t > r \quad q \xmapsto{b(t)} q'}{p\|q \xmapsto{a(r)} p'\|(r \gg q)\ , \quad q\|p \xmapsto{a(r)} (r \gg q)\|p'\ , \quad p \mathbin{\|\!\lfloor} q \xmapsto{a(r)} p' \mathbin{\|\!\lfloor} (r \gg q)}$$

$$\frac{p \xmapsto{a\surd(r)} p' \quad t > r \quad q \xmapsto{b(t)} q'}{p\|q \xmapsto{a(r)} r \gg q\ , \quad q\|p \xmapsto{a(r)} r \gg q\ , \quad p \mathbin{\|\!\lfloor} q \xmapsto{a(r)} r \gg q}$$

$\|, |$:

$$\frac{p \xmapsto{a(r)} p' \quad q \xmapsto{a'(r)} q' \quad a|a' = c \neq \delta}{p\|q \xmapsto{c(r)} p'\|q'\ , \quad p|q \xmapsto{c(r)} p'\|q'} \qquad \frac{p \xmapsto{a\surd(r)} p' \quad q \xmapsto{a'\surd(r)} q' \quad a|a' = c \neq \delta}{p\|q \xmapsto{c\surd(r)} \delta\ , \quad p|q \xmapsto{c\surd(r)} \delta}$$

$$\frac{p \xmapsto{a\surd(r)} p' \quad q \xmapsto{a'(r)} q' \quad a|a' = c \neq \delta}{p\|q \xmapsto{c(r)} q'\ , \quad q\|p \xmapsto{c(r)} q'\ , \quad p|q \xmapsto{c(r)} q'\ , \quad q|p \xmapsto{c(r)} q'}$$

$\gg$:

$$\frac{s > r \quad p \xmapsto{b(r)} p'}{p \gg s \xmapsto{b(r)} p'}$$

δ_H :

$$\frac{p \xmapsto{a(r)} p' \quad a \notin H}{\delta_H(p) \xmapsto{a(r)} \delta_H(p')} \qquad \frac{p \xmapsto{a\surd(r)} p' \quad a \notin H}{\delta_H(p) \xmapsto{a\surd(r)} p'}$$

Table 9: First Part of Transition System Specification for ACPρ

$a, a' \in A_\delta \cup A_{\surd}$

$$\| \;:\; \frac{\left\{ \begin{array}{l} p \xmapsto{a(u)} p' \quad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad p \not\xmapsto{b(r)} \\ q \xmapsto{a'(u)} q' \quad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad q \not\xmapsto{b(r)} \\ \forall c, c' \in A_\delta \cup A_{\surd} \quad (p \not\xmapsto{c(u)} \vee q \not\xmapsto{c'(u)} \vee \hat{c}|\hat{c}' = \delta) \end{array} \right\}}{p \| q \xmapsto{\delta(u)} \delta} \qquad \frac{\left\{ \begin{array}{l} p \xmapsto{\delta(u)} p' \\ t \geq u \quad q \xmapsto{a'(t)} q' \end{array} \right\}}{p \| q \xmapsto{\delta(u)} \delta, \quad q \| p \xmapsto{\delta(u)} \delta}$$

$$| \;:\; \frac{\left\{ \begin{array}{l} p \xmapsto{a(u)} p' \quad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad p \not\xmapsto{b(r)} \\ t \geq u \quad q \xmapsto{a'(t)} q' \\ \forall c, c' \in A_\delta \cup A_{\surd} \quad (p \not\xmapsto{c(u)} \vee q \not\xmapsto{c'(u)} \vee \hat{c}|\hat{c}' = \delta) \end{array} \right\}}{p | q \xmapsto{\delta(u)} \delta, \quad q | p \xmapsto{\delta(u)} \delta}$$

$$\mathbin{\|\!\lfloor} \;:\; \frac{\left\{ \begin{array}{l} q \xmapsto{a(u)} q' \wedge \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad q \not\xmapsto{b(r)} \\ t \geq u \quad p \xmapsto{a'(t)} p' \end{array} \right\}}{p \mathbin{\|\!\lfloor} q \xmapsto{\delta(u)} \delta} \qquad \frac{\left\{ \begin{array}{l} p \xmapsto{\delta(u)} p' \\ t \geq u \quad q \xmapsto{a'(t)} q' \end{array} \right\}}{p \mathbin{\|\!\lfloor} q \xmapsto{\delta(u)} p'}$$

$$\gg \;:\; \frac{\left\{ \begin{array}{l} \forall b \in A_\delta \cup A_{\surd}\, \forall t < s \quad p \not\xmapsto{b(t)} \\ r \geq s \quad p \xmapsto{a(r)} p' \end{array} \right\}}{p \gg s \xmapsto{\delta(s)} \delta}$$

$$\delta_H \;:\; \frac{\left\{ \begin{array}{l} p \xmapsto{a(u)} p' \\ \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad p \not\xmapsto{b(r)} \\ \forall b \in A_\delta \cup A_{\surd} \quad \hat{b} \in A - H \quad p \not\xmapsto{b(u)} \end{array} \right\}}{\delta_H(p) \xmapsto{\delta(u)} \delta}$$

Table 10: Second Part of Transition System Specification for ACPρ in *ntyft* format

4.5 Completeness of ACPρ

Theorem 4.5.1 *Completeness for* ACPρ
$\forall p, q \in T$

$$p \leftrightarrow q \Longrightarrow \text{ACP}\rho \ \vdash p = q$$

Proof. By the soundness of the axioms of ACPρ , Theorem 4.3.1, the Congruence Theorem 4.4.1 and the Elimination Theorem 4.2.3 we may conclude:

$$\exists p_b, q_b \in \mathcal{B} \quad p_b \leftrightarrow p \leftrightarrow q \leftrightarrow q_b$$

By transitivity of bisimulation and the Completeness Theorem 3.5.1 we get

$$\text{ACP}\rho\text{I} \vdash p = q$$

□

5 Integrals

5.1 Introduction

We call *integration* the alternative composition over a continuum of alternatives [BB89].
So, if an action a can happen somewhere in the interval $[1, 2]$ we write:

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

The bounds of an interval may be dependent on *time variables* which are defined in the context.

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

In this section we take a more restrictive view on integration than in [BB89], called *prefixed* integration; we require that every action has as timestamp a *time variable*, directly preceded by the binding integral. Hence, we do not consider "general" integration, integrals over arbitrary (open) terms, as in [BB89]. E.g. we allow the following terms:

$$\int_{v\in V} a(v) \cdot \int_{w\in W} b(w) \quad \text{and} \quad (\int_{v\in V} a(v) + \int_{w\in W} b(w)) \cdot \int_{z\in Z} c(z)$$

but not

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

To prove important facts (like completeness) in detail, we have to introduce some definitions concerning bounds and intervals.

5.2 Some Definitions

- *TVar* is an infinite, countable set of *time variables* . The set of bounds, *Bound*, with typical elements v, v_1, v_2, is defined as follows, where $t \in \mathbb{R}^{\geq 0}_{\omega}, w \in TVar$:

$$v \in Bound \quad v := t \mid w \mid v_1 + v_2 \mid v_1 - v_2 \mid t \cdot v$$

If $v \in Bound$ then the set of *time variables* occurring in v is denoted by $tvar(v)$. We take the freedom to assume the standard notion of identity and the standard ordering on *Bound*, e.g. $1 + 2 = 3$, $v - v = 0$, $v + 2 < v + 3$ but $v + 2 \not< w + 3$. Moreover, we assume $0 < v$ and $v < \omega$ for all $v \in TVar$. Hence, the ordering on *Bound* is not total.

- The set of intervals *Int*:

$$Int = Bool \times Bound \times Bound \times Bool$$

We abbreviate a set $\{V_1, ..., V_n\}$ of intervals by $\overline{V}$. In the sequel $\lhd, \rhd$ are variables ranging over *Bool*. An arbitrary interval $(\lhd, v_0, v_1, \rhd)$ is abbreviated by $\lhd v_0, v_1 \rhd$. Furthermore, we abbreviate

$$\begin{array}{lcl} V_0 = (tt, v_0, v_1, \rhd) & \text{by} & V_0 = [v_0, v_1 \rhd \\ V_1 = (ff, v_0, v_1, \rhd) & \text{by} & V_1 =< v_0, v_1 \rhd \end{array}$$

denoting that V_0 has a closed lower bound and V_1 has an open lower bound. Similarly we abbreviate

$$\begin{array}{lcl} W_0 = (\lhd, v_0, v_1, tt) & \text{by} & W_0 = \lhd v_0, v_1] \\ W_1 = (\lhd, v_0, v_1, ff) & \text{by} & W_1 = \lhd v_0, v_1 > \end{array}$$

denoting that W_0 has a closed upper bound and W_1 has an open upper bound. The set of *time variables* occurring in the *Bound* of an interval V is denoted by $tvar(V)$. Intervals in which no *time variables* occur are called *time-closed intervals.*

$$Int^{time-closed} = Bool \times \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0} \times Bool$$

We use the following abbreviation for $t \in \mathbb{R}^{\geq 0}$ and $V \in Int$

$$\begin{array}{llllll} \text{If} & V =< v_0, v_1 > & \text{then} & t \in V & \text{abbreviates} & v_0 < t < v_1 \\ \text{If} & V = [v_0, v_1 > & \text{then} & t \in V & \text{abbreviates} & v_0 \leq t < v_1 \\ \text{If} & V =< v_0, v_1] & \text{then} & t \in V & \text{abbreviates} & v_0 < t \leq v_1 \\ \text{If} & V = [v_0, v_1] & \text{then} & t \in V & \text{abbreviates} & v_0 \leq t \leq v_1 \end{array}$$

Similarly, $t \notin V$ and $V = \emptyset$ can be defined. Of course we have two (infix) operators $\cap, \cup \in Int^{time-closed} \times Int^{time-closed} \longrightarrow Int^{time-closed}$. The definitions of $\cap$ and $\cup$ are straightforward from the above definition of $t \in V$, with the remark that $V_1 \cup V_2$ is defined if $V_1 \cup V_2$ is again an interval. Furthermore, for $t \in \mathbb{R}^{\geq 0}$ $V \in Int$ we write $t < V$ if t is smaller than every element of V.

$$\begin{array}{llllll} \text{If} & V =< v_0, v_1 \rhd & \text{then} & t < V & \text{abbreviates} & t \leq v_0 \\ \text{If} & V = [v_0, v_1 \rhd & \text{then} & t < V & \text{abbreviates} & t < v_0 \end{array}$$

On *Int* two operators $sup, inf \in Int \longrightarrow Bound$ are defined:
Let $V = \lhd v_0, v_1 \rhd \ \wedge \ v_0, v_1 \in Bound$

$$\begin{array}{rcl} V \neq \emptyset & : & inf(V) = v_0 \\ & : & sup(V) = v_1 \\ V = \emptyset & : & inf(V) = sup(V) = 0 \end{array}$$

Note that all these expressions concerning intervals are in fact abbreviations of a conjunction of linear equations and linear inequalities over *time variables* and numbers from $\mathbb{R}^{\geq 0}$.

- In this section the set of terms over BPA$\rho\delta$I, $\mathcal{T}$, is defined inductively.
  Let $a \in A_\delta$, $V \in Int$

$$
\begin{array}{lcl}
 & & \int_{v \in V}(a(v)) \in \mathcal{T} \\
p \in \mathcal{T} & \Longrightarrow & \int_{v \in V}(a(v) \cdot p) \in \mathcal{T} \\
p, q \in \mathcal{T} & \Longrightarrow & p \cdot q \in \mathcal{T} \\
p, q \in \mathcal{T} & \Longrightarrow & p + q \in \mathcal{T} \\
p \in \mathcal{T}, u \in Bound & \Longrightarrow & u \gg p \in \mathcal{T}
\end{array}
$$

  The *time variable* bound by a first integral of a term p is called a *leading time variable* of p.

- We use the following abbreviation, for $w \in Bound$ and $a \in A_\delta$:

$$\int_{v \in [w,w]} a(v) = a(w)$$

  and again, $\delta(0)$ is abbreviated by δ.

- $FV(p)$ is the set of free *time variables* in p:
  Let $p, q \in \mathcal{T}$, $a \in A_\delta$, $V \in Int$, $u \in Bound$ and $v \in TVar$

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

  In the sequel we will use a *scope convention*, saying that we will not write the scope brackets if the scope is as large as possible. Thus we write $\int_{v \in V} a(v) \cdot p$ for $\int_{v \in V}(a(v) \cdot p)$ if we do not consider this term as the initial part of a sequential composition.

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

$$\mathcal{T}^{time-closed} = \{p \in \mathcal{T} | FV(p) = \emptyset\}$$

- We define a notion of substitution of free *time variables* . If $V \in Int$ then $V[w/v] \in Int$ denotes the interval in which all occurrences of the *time variable* v in the bounds of V are replaced by w. Similarly, we have a substitution on *Bound*, like $u[w/v]$
  Let $p, q \in \mathcal{T}$, $V \in Int$, $a \in A_\delta$, $u \in Bound$, $v, v', v" \in TVar$ where $v \neq v'$ and finally $w \in \mathbb{R}^{>0} \cup TVar$.

$$
\begin{array}{lcl}
(\int_{v" \in V} a(v"))[w/v] & \equiv & \int_{v" \in V[w/v]} a(v") \\
(\int_{v \in V}(a(v) \cdot p))[w/v] & \equiv & \int_{v \in V[w/v]}(a(v) \cdot p) \\
(\int_{v' \in V}(a(v') \cdot p))[w/v] & \equiv & \int_{v' \in V[w/v]} a(v') \cdot (p[w/v]) \\
(p + q)[w/v] & \equiv & p[w/v] + q[w/v] \\
(p \cdot q)[w/v] & \equiv & p[w/v] \cdot q[w/v] \\
(u \gg p)[w/v] & \equiv & u[w/v] \gg p[w/v]
\end{array}
$$

- From now on $\equiv$ denotes syntactic equivalence modulo α-conversion.
  Let $a \in A_\delta$ and $v, w \in TVar$ where $w \notin FV(p)$

$$
\begin{array}{lcl}
\int_{v \in V} a(v) & \equiv & \int_{w \in V} a(w) \\
\int_{v \in V}(a(v) \cdot p) & \equiv & \int_{w \in V}(a(w) \cdot p[w/v])
\end{array}
$$

- We have to adapt the definition of $active(), inittim()$ etc. We give here only the action rules for the atomic cases (including $\int_{v\in V} a(v)\cdot p)$). The other cases do not change.
  Let $p \in \mathcal{T}$, $a \in A$, $V \in Int^{time-closed}$ and $v \in TVar$

$$\begin{array}{lclcl} U(\int_{v\in V}\delta(v)) & = & U(\int_{v\in V}\delta(v)\cdot p) & = & sup(V) \\ U(\int_{v\in V}a(v)) & = & U(\int_{v\in V}(a(v)\cdot p)) & = & sup(V) \\ inittim(\int_{v\in V}a(v)) & = & inittim(\int_{v\in V}(a(v)\cdot p)) & = & V \\ L(X) & = & sup(inittim(X)) & & \end{array}$$

- Since we want to consider $\int_{v\in V'} a(v)$ as a *syntactic summand* of $\int_{v\in V} a(v)$ whenever $V' \subseteq V$, we have to add some rules to our definition of $\sqsubseteq$. Let $p, q \in \mathcal{T}$, $V, V' \in Int^{time-closed}$, $a \in A$ and $v \in TVar$

$$\begin{array}{lcl} V' \subseteq V \ \wedge\ \int_{v\in V} a(v) \sqsubseteq p & \Longrightarrow & \int_{v\in V'} a(v) \sqsubseteq p \\ V' \subseteq V \ \wedge\ \int_{v\in V}(a(v)\cdot q) \sqsubseteq p & \Longrightarrow & \int_{v\in V'}(a(v)\cdot q) \sqsubseteq p \end{array}$$

5.3 The Theory of BPA$\rho\delta$I

In this section we give an axiom system for BPA$\rho\delta$I which can be seen as an extension to and an adaptation of BPA$\rho\delta$ (as given in Table 3). We have four new axioms called INT1-4 which 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 is given in Table 11

INT1	$V = V_1 \cup V_2$	$\int_{v\in V_1} a(v) + \int_{v\in V_2} a(v) = \int_{v\in V} a(v)$
INT2		$\int_{v\in\emptyset} a(v) = \delta$
INT3a	$v \notin FV(Y)$	$\int_{v\in V}(a(v))\cdot Y = \int_{v\in V}(a(v)\cdot Y)$
INT3b	$v \notin FV(Y)$	$\int_{v\in V}(a(v)\cdot X)\cdot Y = \int_{v\in V}(a(v)\cdot(X\cdot Y))$
INT4a		$\{\ \forall t\in V\ :\ X = X + a(v)[t/v]\ \} \implies X = X + \int_{v\in V} a(v)$
INT4b		$\{\ \forall t\in V\ :\ X = X + (a(v)\cdot Y)[t/v]\ \} \implies X = X + \int_{v\in V}(a(v)\cdot Y)$
INT5		$\{\ \forall t\in V\ :\ X[t/v] = Y[t/v]\ \} \implies \int_{v\in V}(a(v)\cdot X) = \int_{v\in V}(a(v)\cdot Y)$
ATI1		$a(0) = \delta$
ATI2		$\int_{v\in V}(\delta(v)\cdot X) = \int_{v\in V}\delta(v)$
ATI3		$\int_{v\in V}\delta(v) = \delta(sup(V))$
ATI4	$t \leq sup(V)$	$\int_{v\in V} a(v) + \delta(t) = \int_{v\in V} a(v)$
ATI5		$\int_{v\in V}(a(v)\cdot X) = \int_{v\in V}(a(v)\cdot(v \gg X))$
ATBI1*a*		$t \gg \int_{v\in V} a(v) = \int_{v\in V\cap<t,\omega>} a(v) + \delta(t)$
ATBI1*b*		$t \gg \int_{v\in V}(a(v)\cdot X) = \int_{v\in V\cap<t,\omega>}(a(v)\cdot X) + \delta(t)$
ATB3		$t \gg (X+Y) = (t\gg X) + (t \gg Y)$
ATB4		$t \gg (X\cdot Y) = (t \gg X)\cdot Y$

Table 11: BPA$\rho\delta$I =BPA+INT1,5+ATI1,5+ATBI1+ATB3,4

INT1 This axiom expresses that an *integration* can be considered as an alternative composition over a continuum of alternatives. Note that terms are only identified in case $V_1 \cup V_2$ is a well defined interval.

INT2 Chosing from the empty set is not defined and corresponds with a deadlock.

INT3 We may widen the scope if the leading *time variable* v of the first part of a sequential composition does not occur in the tail of this sequential composition.

INT4 This axiom is our equivalent of the [BB89] axiom

$$(\forall t \in V \ : \ P(t) = Q(t)) \Longrightarrow \int_{v \in V} P = \int_{v \in V} Q$$

ATI1 Though this axiom looks exactly like ATA1 it is not called ATA1 because $a(0)$ is an abbreviation of $\int_{v \in [0,0]} a(v)$

ATBI1 This is the integral equivalent of both ATB1 and ATB2.

All other axioms are the straightforward equivalents of the BPA$\rho\delta$ axioms.

Removing the Scope Brackets

According to the *scope convention* we can write $\int_{v \in <1,2>}(a(v) \cdot \int_{w \in <3,4>}(b(w)))$ 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.

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

Apply an α-conversion

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

and a *scope widening* according to INT3a,b

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

before removing the brackets according to the *scope convention*

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

In order to make this more formal we need a definition and a Lemma.

Definition 5.3.1 *The set of widest scope terms $\mathcal{W}$ is defined as follows*

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

Lemma 5.3.2 *Every term $p \in \mathcal{T}$ has a widest scope term $p_w \in \mathcal{W}$ such that*

$$INT3 \vdash p = p_w$$

Proof. By induction on the size of p, using *α-conversion* and *scope widening*. □

Remark 5.3.3 *We can write all widest scope term from $\mathcal{W}$ without scope brackets.*

In the following we will not use the scope brackets any more, since we may restrict ourselves to *widest scope* terms.

5.4 Basic Terms

In Definition 2.2.1 the notion of a *basic* term for BPA$\rho\delta$ (without integration) is introduced. In this section we introduce a notion of *basic* terms for BPA$\rho\delta$I. Before giving the formal definition we start with some informal remarks.

Remember that the idea of a *basic* term is, that it does not contain "redundant" information, for example:

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

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. Another characterisation of *basic* terms is, that every subinterval corresponds with a set of transitions in the associated transition system. Or, in other words, for *basic* terms there is a strong correspondence between the intervals and bounds on the one hand, and the set of timestamps of the transition system on the other hand. Indeed, in our completeness proof at the end of this section we will restrict ourselves to *basic* terms, using a Lemma of this section, which states that every term does have a *basic* term.

Definition 5.4.1 *Let $p,q \in \mathcal{W}$, $V \in Int^{time-closed}$, $t \in \mathbb{R}^{\geq 0}$, $r,s \in \mathbb{R}^{\geq 0}_{\omega}$ and $v \in TVar$.*

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

$$\begin{array}{rlcl}
t < V & \int_{v\in V} a(v) & \in & \mathcal{B}^{\omega}(t) \\
t < s & \delta(s) & \in & \mathcal{B}^{s}(t) \\
t < V \ \wedge\ ((\forall r \in V\backslash_{\{\omega\}} \ \ p[r/v] \in \mathcal{B}(r)) \vee p \simeq \delta) & \int_{v\in V}(a(v)\cdot p) & \in & \mathcal{B}^{\omega}(t) \\
p \in \mathcal{B}^{r}(t) \ \wedge\ q \in \mathcal{B}^{s}(t) \ \wedge\ (r = s \ \vee\ (r < s = \omega \ \wedge\ r > U(q))) & p+q, q+p & \in & \mathcal{B}^{r}(t) \\
p \in \mathcal{B}^{r}(t) & p+\delta, \delta+p & \in & \mathcal{B}^{r}(t) \\
 & & & \\
 & \mathcal{B}(t) & = & \bigcup_{r\in\mathbb{R}^{\geq 0}_{\omega}} \mathcal{B}^{r}(t) \\
 & \mathcal{B} & = & \mathcal{B}(0) \cup \{\delta\}
\end{array}$$

Note that $\mathcal{B} \subseteq \mathcal{T}^{time-closed}$. Until now we have dealt only with *time-closed* terms. But if we want to prove by induction that every term has a *basic* term, we have to consider terms with free *time variables* as well. Therefore, we first introduce the notion of a *conditional* term. A *conditional* term is some construct which determines for every 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:

$$\begin{array}{llll}
\textit{if} & \text{the context assigns a value } t \leq 5 \text{ to } v & \textit{then} \text{ deliver} & a(5)\cdot\delta \\
\textit{if} & \text{the context assigns a value } t > 5 \text{ to } v & \textit{then} \text{ deliver} & a(5)\cdot b(t)
\end{array}$$

In the sequel a *conditional* term is denoted as follows (the notation $:\!\longrightarrow$ is taken from [BB90]):

$$p \simeq \begin{array}{l} \{v \leq 5 \ \ :\!\longrightarrow \ \ a(5)\cdot\delta\} \\ + \\ \{v > 5 \ \ :\!\longrightarrow \ \ a(5)\cdot b(v)\} \end{array}$$

For example, the substitution $\sigma \equiv [1/v]$ validates $v \leq 5$, so $\sigma(p) = a(5)\cdot\delta$ which is a *basic* term. Now consider a substitution $\sigma' \equiv [6/v]$. Then σ' validates $v > 5$, so $\sigma'(p) = a(5)\cdot b(6)$ which is again a *basic* term. For every other substitution σ" which assigns a real number to v we have $\sigma"(p) \in \mathcal{B}$. In the following we introduce a generalisation of a *basic* term, called a *conditional basic* term. The idea is that a *conditional* term is called *basic* if any substitution of reals for the free *time variables* applied on the *conditional* term gives a *basic* term.

In the sequel we introduce a machinery by which we can state that the term $w \gg \int_{v\in<v_0,v_1>} a(v)$ with $w, v_0, v_1 \in TVar$ has the following *conditional basic* term z associated with it:

$$z \simeq \begin{array}{lll} \{v_0 \geq v_1 \ \vee \ w \geq v_1 & :\longrightarrow & \delta(w)\} \\ + & & \\ \{v_0 < w < v_1 & :\longrightarrow & \int_{v\in<w,v_1>} a(v)\} \\ + & & \\ \{w \leq v_0 < v_1 & :\longrightarrow & \int_{v\in<v_0,v_1>} a(v)\} \end{array}$$

or, if we take $V =< v_0, v_1 >$

$$z \simeq \begin{array}{lll} \{w \geq sup(V) & :\longrightarrow & \delta(w)\} \\ + & & \\ \{w < sup(V) & :\longrightarrow & \int_{v\in V\cap<w,\omega>} a(v)\} \end{array}$$

Logical expressions like $w \leq v_0 < v_1$ are called *conditions.* Every *condition* represents a set of substitutions. In the example above there are three *conditions*, which together form a partition. Consider a substitution σ with $\sigma(v_0) = 1$, $\sigma(w) = 2$ and $\sigma(v_1) = 3$. Then the second *condition*, $v_0 < w < v_1$, is validated by σ (later we will say that $\sigma \in [v_0 < w < v_1]$). The associated term to this *condition* on the leftside of the $:\longrightarrow$ is $\int_{v\in<w,v_1>} a(v)$, e.g.

$$\sigma(z) \simeq \sigma(\begin{array}{lll} \{1 > 2 \ \vee \ 2 \leq 1 & :\longrightarrow & \delta(w)\} \\ + & & \\ \{1 < 2 < 3 & :\longrightarrow & \int_{v\in<2,3>} a(v)\} \\ + & & \\ \{2 \leq 1 < 3 & :\longrightarrow & \int_{v\in<1,3>} a(v)\} \end{array}) \overset{1<2\leq 3=tt}{\simeq} \int_{v\in<2,3>} a(v) \in \mathcal{B}\,(2)$$

$$\Longrightarrow \quad \sigma(z) \in \mathcal{B}\,(\sigma(w))$$

Conditional Terms

All these new notions are defined more formally in the following definitions:

- A substitution $\sigma \in \Sigma$ is a function which assigns to each *time variable* a real number or the *time variable* itself. On $\mathbb{R}^{\geq 0}$ each substitution σ behaves as the identity. If $\sigma \in \Sigma$, then we define $\sigma \in \mathcal{T} \longrightarrow \mathcal{T}$ by induction on the *size* of p. Moreover, we define $\sigma \ : \ Int \longrightarrow Int.$

$$\begin{array}{lll} \Sigma^* & = & (TVar \cup \mathbb{R}^{\geq 0}) \longrightarrow^{tot} (TVar \cup \mathbb{R}^{\geq 0}) \\ \Sigma & = & \{\sigma \in \Sigma^* \mid \forall t \in \mathbb{R}^{\geq 0} \quad \sigma(t) = t \ \wedge \\ & & \quad \forall v \in TVar \quad \sigma(v) > 0 \ \wedge \ (\sigma(v) \in TVar \Longrightarrow \sigma(v) = v)\} \\ \sigma_{\backslash v} & = & \lambda w.if \ v = w \ then \ v \ else \ \sigma(w) \\ \sigma(< v_0, v_1]) & = & < \sigma(v_0), \sigma(v_1)] \\ \sigma(\int_{v\in V} a(v) \cdot p) & = & \int_{v\in\sigma(V)} a(v) \cdot \sigma_{\backslash v}(p) \end{array}$$

- Some subsets of Σ may be represented by a *condition.* The set $Cond^{at}$ of *atomic conditions* is as follows:

$$Cond^{at} \ = \ \{v_0 < v_1, v_0 = v_1 | v_0, v_1 \in Bound\} \cup \{tt, ff\}$$

  An *atomic condition* is a linear equation or inequality over *time variables* and real numbers. Let $tvar(\alpha)$ denote the set of *time variable* occurring in α. Hence, if α is an equation from $Cond^{at}$ and $v \in tvar(\alpha)$ then we can write α as $v = w$ for some $w \in Bound.$

- The set *conditions* consists of *atomic conditions* combined by logical connectives (α^{at} is taken from $Cond^{at}$):

$$\alpha \in Cond \quad \alpha := \alpha^{at} \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \neg(\alpha)$$

In *Cond* we distinguish a subset $Cond^n$, the *conditions* in normal form. A *condition* is in normal form if it is a conjunction of *atomic conditions.* We denote the set of *time variables* of $\alpha \in Cond$ by $tvar(\alpha)$. We abbreviate a set of *conditions* $\{\alpha_1, ..., \alpha_n\}$ by $\overline{\alpha}$. Similarly, $tvar(\overline{\alpha}$ is an abbreviation for $tvar(\alpha_1) \cup ... \cup tvar(\alpha_n)$.

- Every $\alpha \in Cond$ stands for a subset of substitutions for the *time variables* of $TVar$, so:

$$\begin{array}{lcl} [tt] & = & \Sigma \\ [ff] & = & \emptyset \\ [v < w] & = & \{\sigma \in \Sigma \mid \sigma(v) < \sigma(w)\} \\ [v = w] & = & \{\sigma \in \Sigma \mid \sigma(v) = \sigma(w)\} \end{array}$$
$$[\alpha \vee \beta] = [\alpha] \cup [\beta] \text{ and } [\alpha \wedge \beta] = [\alpha] \cap [\beta] \text{ and } [\neg(\alpha)] = \Sigma - [\alpha]$$

Earlier we introduced a notion of equality and an ordering on *Bound.* For example, we have $v + 1 > v$ and $w = w$ but $w + 1 \neq v$. Hence, $[v + 1 > v] = \Sigma = [tt]$. Moreover, if $\alpha \in Cond^{at}$ and $[\alpha] \neq [tt]$ then we have for all $\sigma \in [\alpha]$ that $\sigma(tvar(\alpha)) \subset \mathbb{R}^{>0}$.

- A set of *conditions* like $\{\alpha_1, ..., \alpha_n\}$ is called a partition if for every $\sigma \in \Sigma$ with $\sigma(tvar(\overline{\alpha})) \subset \mathbb{R}^{>0}$ we have $\exists! i \quad \sigma \in [\alpha_i]$.

- $\mathcal{C}$, with typical elements p_c, is the set of conditional terms.

  Let $p \in \mathcal{W}$, $\alpha \in Cond$ and $v \in Bound$

$$p_c \in \mathcal{C} \quad p_c := p \mid \alpha :\longrightarrow p_c \mid p_c + p_c \mid \int_{w \in W} (a(w) \cdot p_c) \mid v \gg p_c$$

We may abbreviate a set $\{p_1, ..., p_n\}$ of terms from $\mathcal{C}$ by $\overline{p}$. If $\sigma \in \Sigma$ then we define $\sigma \ : \ \mathcal{C} \longrightarrow \mathcal{C}$ as expected. We have

$$\sigma(\alpha :\longrightarrow p) = \begin{cases} \sigma(p) & \text{if } \sigma \in [\alpha] \\ \delta & \text{otherwise} \end{cases}$$

On $\mathcal{C}$ the equivalence $\simeq$ means syntactic equivalence modulo α-conversion, associativity and commutativity of the $+$ and modulo equivalence on *conditions* . Hence, if $[\alpha] = [\beta]$ then $\alpha :\longrightarrow p \simeq \beta :\longrightarrow p$. Equivalences on $\mathcal{C}$ may be hard to read, e.g. $v = w \longrightarrow p = w = v \longrightarrow p$. Therefore, we may write *conditional terms* between brackets. For example $\{v = w \longrightarrow p\} = \{w = v \longrightarrow p\}$. These brackets are not part of the syntax but are meta-brackets.

- Instead of $\{v < w :\longrightarrow p + v\} = \{w :\longrightarrow p\}$ we write $\{v \leq w :\longrightarrow p\}$. Furthermore, we denote $\{v_0 < v \leq v_1 :\longrightarrow p\}$ by $\{v \in \, < v_0, v_1] :\longrightarrow p\}$. Hence, *conditions* may contain subconditions of the form $v \in V$ for arbitrary $v \in Bound, V \in Int$. Thus, if $v \in TVar$ and $V \in Int$ then the *atomic condition* $v \in V$ must be read as "the set γ of substitutions such that for each σ in γ we have $\sigma(v) \in \sigma(V)$".

- In Table 12 we state identities over $\mathcal{C}$, resulting in an equational theory BPA$\rho\delta$IC if combined with the axioms of BPA$\rho\delta$I. A substitution can be seen as an abstraction of a context. We define the set on contexts, *Context*, where the empty context is denoted by [] and p is taken from $\mathcal{T}$.

$$C[] \in Context \quad C[] := [] \mid C[] + p \mid p + C[] \mid \int_{v \in V} (a(v) \cdot C[]) \mid u \gg C[]$$

$Context^{time-closed}$ is the set containing all *time-closed* contexts. Every $C[]$ has exactly one occurrence of $[]$, which may be in the scope of some integrals binding *time variables* , which are called the *bound* variables of $C[]$. Thus, $BV([]) = \emptyset$ and $BV(\int_{v\in V}(a(v) \cdot C[])) = \{v\} \cup BV(C[])$. For every $p \in \mathcal{T}$, and $C[] \in Context$ we can construct a term $C[p] \in \mathcal{T}$. Note that $[p] \equiv p$.

COND1	$\{tt :\rightarrow p\} = p$
COND2	$\{ff :\rightarrow p\} = \delta$
COND3	$\{\alpha :\rightarrow (\beta :\rightarrow p)\} = \{\alpha \wedge \beta :\rightarrow p\}$
COND4	$\{\alpha :\rightarrow p + q\} = \{\alpha :\rightarrow p\} + \{\alpha :\rightarrow q\}$
COND5	$\{\alpha \vee \beta :\rightarrow p\} = \{\alpha :\rightarrow p\} + \{\beta :\rightarrow p\}$
COND6	$r \gg \{\alpha :\rightarrow p\} = \{\alpha :\rightarrow r \gg p\}$
COND7	$[\alpha] = [\beta] \quad \Longrightarrow \quad \{\alpha :\rightarrow p\} = \{\beta :\rightarrow p\}$
COND8a	$\{cl \geq sup(V) \ :\rightarrow \ cl \gg \int_{v\in V} a(v)\} = \{cl \geq sup(V) \ :\rightarrow \ \delta(cl)\}$
COND8b	$\{cl \geq sup(V) \ :\rightarrow \ cl \gg \int_{v\in V} a(v) \cdot p\} = \{cl \geq sup(V) \ :\rightarrow \ \delta(cl)\}$
	$V_{cl} = V \cap < cl, \omega >$
COND9a	$\{cl < sup(V) \ :\rightarrow \ cl \gg \int_{v\in V} a(v)\} = \{cl < sup(V) \ :\rightarrow \ \int_{v\in V_{cl}} a(v)\}$
COND9b	$\{cl < sup(V) \ :\rightarrow \ cl \gg \int_{v\in V} a(v) \cdot p\} = \{cl < sup(V) \ :\rightarrow \ \int_{v\in V_{cl}} a(v) \cdot p\}$
COND10a	$\{V = \emptyset \ :\rightarrow \ \int_{v\in V} a(v)\} = \{V = \emptyset \ :\rightarrow \ \delta)\}$
COND10b	$\{V = \emptyset \ :\rightarrow \ \int_{v\in V} a(v) \cdot p\} = \{V = \emptyset \ :\rightarrow \ \delta\}$
COND11	If $(\alpha_1 \wedge v \in W_1), ..., (\alpha_{\#I} \wedge v \in W_{\#I})$ is a partition and $v \notin tvar(\overline{\alpha})$ 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 12: BPA$\rho\delta$IC = BPA$\rho\delta$I + COND 1-7 + COND 8-11

The justification of these conditional axioms is given by the following Lemma. Note that it is possible now to reason with terms, even if they contain free *time variables*, by using these conditional axioms. Moreover, this Lemma implies that all the conditional axioms together have the same expressive power as the axiom INT5 (as long as we restrict ourselves to *prefixed* integration).

Lemma 5.4.2 $\forall p, q \in \mathcal{T}$

BPA$\rho\delta$IC $\vdash p = q \iff \forall C[] \in Context^{time-closed}$ *with* $FV(p{+}q) \subseteq BV(C[])$ BPA$\rho\delta$I $\vdash C[p] = C[q]$

Proof. By induction on the depth of $[]$ in $C[]$ using $INT5$. □

In order to use COND11 we need the following Lemma which states that is possible to isolate all occurrences of a variable from $TVar$ in α, by rewriting α into one or several *conditions* of the form

$\beta_i \wedge v \in V_i$. For example, using the next Lemma we can derive that

$$p \equiv \begin{array}{l} \{v < 3 \wedge v < w \wedge 1 < v \quad :\rightarrow \quad q\} \\ + \\ \{3 < v \wedge v = w \wedge v < z \quad :\rightarrow \quad q'\} \end{array} = \begin{array}{ll} \{3 \leq w \wedge v \in < 1, 3 > & :\rightarrow \quad q\} \\ + & \\ \{w \leq 3 \wedge v \in < 1, w > & :\rightarrow \quad q\} \\ + & \\ \{3 < w \wedge w < z \wedge v \in [w, w] & :\rightarrow \quad q'\} \end{array}$$

Now, if we consider the term $\int_{v \in < w, 10 >} a(v) \cdot p$ we can "push" the *conditions* upwards by applying COND11.

Lemma 5.4.3 $\forall \alpha \in Cond^n \; \forall p \in C \; \forall v \in TVar \quad \exists \overline{\beta} \subset Cond^n$ *with* $v \notin tvar(\overline{\beta}) \; \exists \overline{V} \subseteq Int$

$$\text{BPA}\rho\delta\text{IC} \vdash \{\alpha :\rightarrow p\} = \sum_j \{\beta_j \wedge v \in V_j \; :\rightarrow \; p\}$$

Proof. We denote that part of α in which v does not occur by $\alpha_{\backslash v}$. We abbreviate $\alpha_1 \wedge \ldots \wedge \alpha_n$ by $\bigwedge_{i \in \{1,\ldots,n\}} \alpha_i$, where $\bigwedge_{i \in \emptyset} \alpha_i = tt$.

- $v \notin tvar(\alpha)$

$$\{\alpha \; :\rightarrow \; p\} \simeq \{\alpha \wedge v \in \mathbb{R}^{>0} \; :\rightarrow \; p\}$$

- v occurs in α in a *simple condition* of the form $v = c$ for some $c \in Bound$.

  $\alpha = \{ \alpha_{\backslash v} \wedge \bigwedge_{i \in I} v > a_i \wedge \bigwedge_{j \in J} v < b_j \wedge \bigwedge_{k \in K} v = w_k \}$ for some I, J and $K, K \neq \emptyset$
  We take $c = w_k$ for some w_k and substitute
  $\alpha = \{ \alpha_{\backslash v} \wedge \bigwedge_{i \in I} c > a_i \wedge \bigwedge_{j \in J} c < b_j \wedge \bigwedge_{k \in K} c = w_k \wedge v = c \}$
  thus for some β in which v does not occur we have
  $\alpha = \{ \beta \wedge v = c \}$

  Hence,

$$\{\alpha \; :\rightarrow \; p\} \simeq \{\beta \wedge v \in [c, c] \; :\rightarrow \; p\}$$

- v occurs in α only in inequalities.

  $\alpha = \{ \alpha_{\backslash v} \wedge \bigwedge_{i \in I} v > a_i \wedge \bigwedge_{j \in J} v < b_j \}$
  for some I, J with $I \cup J \neq \emptyset$. This *condition* can be rewritten as follows
  $\alpha = \bigvee_{i,j \in I \times J} \{ \alpha_{\backslash v} \wedge \bigwedge_{i' \in I} a_i \geq a_{i'} \wedge \bigwedge_{j' \in J} b_j \leq b_{j'} \wedge v > a_i \wedge v < b_j \}$
  Thus there are $\gamma_{i,j}$ in which v does not occur, such that
  $\alpha = \bigvee_{i,j \in I \times J} \{ \gamma_{i,j} \wedge v > a_i \wedge v < b_j \}$

$$\{\alpha \; :\rightarrow \; p\} = \sum_{i,j \in I \times J} \{\gamma_{i,j} \wedge v \in < a_i, b_j > :\rightarrow \; p\}$$

□

Corollary 5.4.4 $\forall p \in C \; \forall v \in TVar$

$$p \simeq \textstyle\sum_i \{\alpha_i \; :\rightarrow \; p_i\}$$
$$\Longrightarrow$$
$$\exists \overline{\beta} \subset Cond \; v \notin tvar(\overline{\beta}) \; \exists \overline{V} \subset Int \; \exists \overline{q} \subset C \quad p \simeq \textstyle\sum_j \{\beta_j \wedge v \in V_j \; :\rightarrow \; q_j\}$$

We define a notion of *basic conditional* terms:

Definition 5.4.5 $v \in Bound$

$$\sum_i \{\alpha_i :\to p_i\} \in \mathcal{B}(v) \quad \overset{def}{\Longleftrightarrow} \quad \begin{cases} \overline{\alpha} \textit{ is a partition} \\ \wedge \\ \forall \sigma \quad \sigma \in [\alpha_i] \Longrightarrow \sigma(p_i) \in \mathcal{B}(\sigma(v)) \cup \{\delta\} \end{cases}$$

For example $\int_{v\in<z,z+1>} a(v)\cdot\int_{w\in<v,\omega>} b(w) \in \mathcal{B}(z)$ but $\int_{v\in<z,10>} a(v) \notin \mathcal{B}(z)$. Note that $\int_{v\in<z,10>} a(v)$ can be rewritten to the (*conditional*) *basic* term $\{z < 10 :\to \int_{v\in<z,10>} a(v)\} + \{z \geq 10 :\to \delta(cl)\}$. The next Lemma states that every term in $\mathcal{T}$ (with possibly free *time variables*) can be rewritten to a *basic* (*conditional*) term.

Lemma 5.4.6 $\forall p \in \mathcal{C}\ \forall cl \in Bound \quad : \quad \exists b_{p,cl} \in \mathcal{B}(cl)$ *with*

$$FV(b_{p,cl}) \subseteq FV(cl \gg p) \quad \wedge \quad \mathrm{BPA}\rho\delta\mathrm{IC} \vdash cl \gg p = b_{p,cl}$$

Proof. By induction on the *size* of p. Again, V_{cl} is an abbreviation of $V \cap < cl, \omega >$. Note that it would not be possible to give this proof if we would integrate over arbitrary subsets of $\mathbb{R}^{\geq 0}$, as is the case in general integration.

- $\int_{v\in V} a(v)$

$$\begin{aligned} & cl \gg \textstyle\int_{v\in V} a(v) \\ = \quad & \{cl \geq sup(V) :\to \delta(cl)\} \ + \ \{cl < sup(V) :\to \textstyle\int_{v\in V_{cl}} a(v)\} \end{aligned}$$

- $p + q$

$$\begin{aligned} & cl \gg (p+q) \\ = \quad & (cl \gg p) + (cl \gg q) \\ = \quad & \textstyle\sum_i\{\ \alpha_i :\to p_i\ \} \ + \ \sum_j\{\ \beta_j :\to q_j\ \} \quad \text{by induction} \\ = \quad & \textstyle\sum_{(i,j)}\{\ \alpha_i \ \wedge\ \beta_j \ :\to \ p_i + q_j\ \} \quad \text{since } \overline{\alpha} \text{ and } \overline{\beta} \text{ are partitions} \\ = \quad & \textstyle\sum_{(i,j)} \begin{Bmatrix} \alpha_i \ \wedge\ \beta_j \ \wedge\ U(p_i+q_j) > U(\overline{p_i}+\overline{q_j}) & :\to & p_i + q_j \\ + & & \\ \alpha_i \ \wedge\ \beta_j \ \wedge\ U(p_i+q_j) \leq U(\overline{p_i}+\overline{q_j}) & :\to & p_i + q_j \end{Bmatrix} \\ & \text{where } \overline{p} \text{ is the term } p \text{ without } \delta\text{-summands} \\ = \quad & \textstyle\sum_{(i,j)} \begin{Bmatrix} \alpha_i \ \wedge\ \beta_j \ \wedge\ U(p_i+q_j) > U(\overline{p_i}+\overline{q_j}) & :\to & \overline{p_i}+\overline{q_j}+\delta(U(p_i+q_j)) \\ + & & \\ \alpha_i \ \wedge\ \beta_j \ \wedge\ U(p_i+q_j) \leq U(\overline{p_i}+\overline{q_j}) & :\to & \overline{p_i}+\overline{q_j} \end{Bmatrix} \end{aligned}$$

- $\int_{v \in V} a(v) \cdot p$

$$
\begin{array}{ll}
 & cl \gg \int_{v \in V} a(v) \cdot p \\
= & \{cl \geq sup(V) :\rightarrow \delta(cl)\} \ + \ \{cl < sup(V) :\rightarrow \int_{v \in V_{cl}} (a(v) \cdot (v \gg p))\} \\
= & \{cl \geq sup(V) :\rightarrow \delta(cl)\} \ + \ \{cl < sup(V) :\rightarrow \int_{v \in V_{cl}} (a(v) \cdot \sum_i \{\alpha_i \wedge v \in W_i :\rightarrow p_i\})\} \\
 & \begin{array}{lll}
\{cl \geq sup(V) & :\rightarrow & \delta(cl)\} \\
+ & & \\
\{\sum_i \{ \ cl < sup(V) \wedge \alpha_i \ \wedge \ V_{cl} \cap W_i = \emptyset\} & :\rightarrow & \delta(cl)\} \\
+ & & \\
\{\sum_i \{ \ cl < sup(V) \wedge \alpha_i \ \wedge \ V_{cl} \cap W_i \neq \emptyset & :\rightarrow & \int_{v \in V_{cl} \cap W_i} a(v) \cdot p_i \ \}\}
\end{array}
\end{array}
$$

Some of these p_i's may be of the form $\delta(v)$ and must be rewritten to δ. Since we have to end up with a *basic* term we have to construct a partition. This can be done by applying sufficiently many times the rules

$$
\begin{array}{lll}
\{\alpha & :\rightarrow & p\} \\
+ & & \\
\{\beta & :\rightarrow & q\}
\end{array}
\longrightarrow
\begin{array}{lll}
\{\alpha \wedge \beta & :\rightarrow & p + q\} \\
+ & & \\
\{\alpha \wedge \neg(\beta) & :\rightarrow & p\} \\
+ & & \\
\{\neg(\alpha) \wedge \beta & :\rightarrow & q\}
\end{array}
$$

and

$$
\{\alpha :\rightarrow p + \delta(t)\} \longrightarrow \{\alpha \ \wedge \ t > U(p) :\rightarrow p + \delta(t)\} + \{\alpha \ \wedge \ t \leq U(p) :\rightarrow p\}
$$

- $r \gg p$

$$
\begin{array}{ll}
 & cl \gg (r \gg p) \\
= & cl \gg \sum_i \{\alpha_i :\rightarrow p_i\} \quad \text{by induction} \\
= & \sum_i \{\alpha_i :\rightarrow cl \gg p_i\} \\
= & \sum_i \{\alpha_i :\rightarrow \sum_j \{\beta_j :\rightarrow p_{i,j}\}\} \\
= & \sum_{i,j} \{(\alpha_i \wedge \beta_j) :\rightarrow p_{i,j}\}
\end{array}
$$

□

Lemma 5.4.7 $\forall p \in \mathcal{T} \ \exists p_b \in \mathcal{C}$ *with* $FV(p_b) \subseteq FV(p) \ \wedge \ p_b \in \mathcal{B}$ *such that*

$$\mathrm{BPA}\rho\delta\mathrm{IC} \vdash p = p_b$$

Proof. Take $p_b = b_{p,0}$ □

Theorem 5.4.8 $\forall p \in \mathcal{T}^{time-closed} \ \exists p_b \in B$ *such that*

$$\mathrm{BPA}\rho\delta\mathrm{I} \vdash p = p_b$$

Proof. According to the previous Lemma there is a $p_b \in \mathcal{B}(0)$ with $FV(p_b) \subseteq FV(p) = \emptyset$ and BPA$\rho\delta$IC $\vdash p = (0 \gg p) = p_b$. By Lemma 5.4.2 and $FV(p + p_b) = \emptyset$ we know that for every *context* $C[]$ we have BPA$\rho\delta$I $\vdash C[p] = C[p_b]$. So we can take the empty context $[]$ and the Theorem follows by $[q] \equiv q$ for all $q \in \mathcal{T}$. □

5.5 An Abstract Operational Semantics for BPA$\rho\delta$I

In order to obtain action rules for BPA$\rho\delta$I, we have to adapt the action rules for the atomic cases for BPA$\rho\delta$. The other action rules can be taken from Table 5.

We expect the following transitions:

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

since $5 \in <0,10>$. The Transition System Specification for BPA$\rho\delta$I is given in Table 13. Bisimulation equivalence is denoted by $\underline{\leftrightarrow}$ and its definition can be found in Definition 3.3.1.

Import rules w.r.t. $\cdot$ + and *abs. shift* from Table 4
$a \in A$ *atom* : $r \in V\backslash\{0\}$ $\int_{v \in V} a(v) \xrightarrow{a(r)} \surd$ $r \in V\backslash\{0\}$ $\int_{v \in V}(a(v) \cdot p) \xrightarrow{a(r)} r \gg p[r/v]$
$\dfrac{U(p) > L(p)}{p \xrightarrow{\delta(U(p))} \surd}$

Table 13: Transition System Specification for BPA$\rho\delta$I

Theorem 5.5.1 *The axioms of* BPA$\rho\delta$I *are sound with respect to bisimulation equivalence.*

Proof. Straightforward □

Theorem 5.5.2 *The operational semantics presented in this section for* BPA$\rho\delta$I *is equivalent with the original semantics for* BPA$\rho\delta$I*:*
$\forall p, q \in \mathcal{T}^{time-closed}$

$$p \underline{\leftrightarrow} q \Longleftrightarrow <p,0> \underline{\leftrightarrow} <q,0>$$

Proof. Straightforward □

The correspondence between the syntax of a *basic* term and its transition system can be stated an proved as the *Correspondence Lemma* 3.3.4. Moreover, we have a Lemma similar to 3.3.5 stating that *basic* terms have ascending timestamps.

5.6 An Abstract Operational Semantics for BPA$\rho\delta$I in *ntyft* format

Now we have to introduce action rules in *ntyft* format which are equivalent with the action rules in Table 13. The rules for the atomic cases are straightforward translations of the *atom* rules of the Table 13. We can import the action rules in *ntyft* format w.r.t. $\cdot$ and *(absolute) time shift* from Table 4, where the action rules for BPA$\rho\delta$ are given. However, we cannot take the action rule for the + of Table 4, since we expect:

$$\int_{v\in<0,1>} a(v)+\delta(1) \overset{\delta(1)}{\not\longmapsto}$$

because every *idle* step contributed by $\delta(1)$ is contributed by $\int_{v\in<0,1>} a(v)$ as well.

In Table 14 a Transition System Specification in *ntyft* format is given. The transition relation is denoted by $\longmapsto$. Again we denote bisimulation equivalence for $\longmapsto$ by $\underline{\leftrightarrow}_{\longmapsto}$. Note that

$$\frac{U(p+q)>L(p+q)}{p+q \overset{\delta(U(p+q))}{\longmapsto} \delta} \quad \text{corresponds to the rule} \quad \frac{\{\ U(p)>L(p) \quad U(p)>L(q) \quad U(p)\geq U(q)\ \}}{p+q \overset{\delta(U(p))}{\longmapsto} \delta,\quad q+p \overset{\delta(U(p))}{\longmapsto}}$$

And this last rule corresponds in its turn to the δ rule for the + of Table 14.

Import rules w.r.t. $\cdot$ +, and *abs. shift* from Table 6, except the δ-rule for the +
$a\in A$ *atom* : $r\in V \quad \int_{v\in V} a(v) \overset{a\surd(r)}{\longmapsto} \delta$ $r\in V \quad \int_{v\in V}(a(v)\cdot p) \overset{a(r)}{\longmapsto} r\gg p[r/v]$
$sup(V)\neq 0$ *atom* : $\int_{v\in V} \delta(v) \overset{\delta(sup(V))}{\longrightarrow} \delta$ + : $\dfrac{\{\ p \overset{\delta(u)}{\longmapsto} p' \quad r<u \quad \forall r'\geq r\, \forall b\in A \quad q \overset{b(r')}{\not\longmapsto} \quad \forall t>u \quad q \overset{\delta(t)}{\not\longmapsto}\ \}}{p+q \overset{\delta(u)}{\longmapsto} p',\quad q+p \overset{\delta(u)}{\longmapsto} p'}$

Table 14: Transition System Specification for BPA$\rho\delta$I in *ntyft* format

Theorem 5.6.1 *Bisimulation is a congruence*

Proof. See the proof of Theorem 3.4.2. The above rules are again in *ntyft* format and the stratification of the proof of 3.4.2 remains valid □

Lemma 5.6.2 $\forall p \in \mathcal{T}^{time-closed}\ \forall u \in \mathbb{R}^{\geq 0}$

$$u = U(p) > L(p) \iff p \overset{\delta(u)}{\longmapsto} \delta$$

Proof. See Appendix C □

5.7 Completeness of BPA$\rho\delta$I

Theorem 5.7.1 *Completeness*
$\forall p, q \in \mathcal{T}$

$$p \leftrightarrow q \Longrightarrow \text{BPA}\rho\delta\text{I} \vdash p = q$$

We may restrict ourselves to *basic* terms, since every term is derivably equal to a *basic* term and since we have already proved soundness of BPA$\rho\delta$I .
Proof. We prove that $p \subseteq q$ by induction on the *size* of p. By symmetry, $q \subseteq p$ follows and we are done. Assume

$$p \simeq \sum_i \int_{v_i \in V_i} a_i(v_i) \cdot p_i + \sum_j \int_{w_j \in W_j} b_j(w_j)$$

$\forall i$	$\forall t \in V_i$	$p \xrightarrow{a_i(t)} t \gg p_i[t/v_i]$	By bisimulation q also has a transition
	$\exists q'$	$q \xrightarrow{a_i(t)} t \gg q'$	where
		$t \gg p_i[t/v_i] \leftrightarrow t \gg q'$	Since both p and q are *basic* terms
		$p_i[t/v_i] \leftrightarrow t \gg p_i[t/v_i]$	and
		$t \gg q' \leftrightarrow q'$	and by induction and the transitivity of $\leftrightarrow$
		$p_i[t/v_i] = q'$	Together with
		$a_i(t) \cdot q' \sqsubseteq q$	we may conclude
		$a_i(t) \cdot p_i[t/v_i] \subseteq q$	
	$(\overrightarrow{INT}4)$	$\int_{v \in V_i} a(v) \cdot p_i \subseteq q$	

$\forall j$	$\forall t \in W_j$	$p \xrightarrow{b_j(t)} \surd$	By bisimulation also q has a transition
		$q \xrightarrow{b_j(t)} \surd$	and thus
		$b_j(t) \sqsubseteq q$	
	$(\overrightarrow{INT}4)$	$\int_{w \in W_j} b_j(w) \subseteq q$	

adding together : $p \subseteq q$

□

5.8 The Theory of ACPρI

In this section we incorporate the operators $\|, |$ and $\lfloor\!\lfloor$ 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

$$\begin{array}{lll} p, q \in \mathcal{T}, \square \in \{\|, |, \lfloor\!\lfloor\} & \Longrightarrow & p \square q \in \mathcal{T} \\ s \in \mathbb{R}^{\geq 0} & \Longrightarrow & p \gg s \in \mathcal{T} \end{array}$$

to the definition of $\mathcal{T}$ in Section 5.2. Next, the definitions of $FV()$ and substitution must be adapted. Moreover, we have to extend the definition of the syntactic operators $initact(), inittim()$ (see 4.1.2). We give only the new rules for the atomic cases, the other rules from 4.1.2 remain valid.

Let $V = Int^{time-closed}$

$$\begin{array}{lllll} initact(\int_{v \in V} \delta(v)) & = & initact(\int_{v \in V} \delta(v) \cdot X) & = & \emptyset \\ initact(\int_{v \in V} a(v)) & = & initact(\int_{v \in V}(a(v) \cdot X)) & = & \{a(t) | t \in V \backslash_{\{0\}}\} \end{array}$$

We introduce a theory ACPρI, which consists of adapted axioms of ACPρ as given in Table 7.

$a, b \in A_\delta$ $V \in Int^{time-closed}$

ATBI5a	$r > inf(V)$	$\int_{v \in V} a(v) \gg r = \int_{v \in V \cap <0,r>} a(v)$
ATBI5b	$r > inf(V)$	$\int_{v \in V}(a(v) \cdot p) \gg r = \int_{v \in V \cap <0,r>}(a(v) \cdot p)$
ATBI6a	$r \leq inf(V)$	$\int_{v \in V} a(v) \gg r = \delta(r)$
ATBI6b	$r \leq inf(V)$	$\int_{v \in V}(a(v) \cdot p) \gg r = \delta(r)$
ATB7		$(X+Y) \gg t = (X \gg t) + (Y \gg t)$
ATB8		$(X \cdot Y) \gg t = (X \gg t) \cdot Y$
ATCI1	$V_0 \cap V_1 \neq \emptyset$	$\int_{v \in V_0} a(v) \| \int_{v \in V_1} b(v) = \int_{v \in V_0 \cap V_1}(a\|b)(v)$
ATCI2	$V_0 \cap V_1 = \emptyset$	$\int_{v \in V_0} a(v) \| \int_{v \in V_1} b(v) = \delta(min(sup(V_0), sup(V_1)))$
CM1		$X \| Y = X \lfloor\!\lfloor Y + Y \lfloor\!\lfloor X + X \| Y$
ATCMI2	$v \notin FV(Y)$	$\int_{v \in V}(a(v)) \lfloor\!\lfloor Y = \int_{v \in V}(a(v) \gg U(Y)) \cdot Y$
ATCMI3	$v \notin FV(Y)$	$\int_{v \in V}(a(v) \cdot X) \lfloor\!\lfloor Y = \int_{v \in V}((a(v) \gg U(Y)) \cdot (X \| Y))$
CM4		$(X+Y) \lfloor\!\lfloor Z = X \lfloor\!\lfloor Y + Y \lfloor\!\lfloor Z$
CMI5	$V_0 \cap V_1 \neq \emptyset$	$\int_{v \in V_0}(a(v) \cdot X) \| \int_{v \in V_1} b(v) = \int_{v \in V_0 \cap V_1}((a\|b)(v) \cdot X)$
CMI5$'$	$V_0 \cap V_1 = \emptyset$	$\int_{v \in V_0}(a(v) \cdot X) \| \int_{v \in V_1} b(v) = \delta(min(sup(V_0), sup(V_1)))$
CMI6	$V_0 \cap V_1 \neq \emptyset$	$\int_{v \in V_0} a(v) \| \int_{v \in V_1}(b(v) \cdot Y) = \int_{v \in V_0 \cap V_1}((a\|b)(v) \cdot Y)$
CMI6$'$	$V_0 \cap V_1 = \emptyset$	$\int_{v \in V_0} a(v) \| \int_{v \in V_1}(b(v) \cdot Y) = \delta(min(sup(V_0), sup(V_1)))$
CMI7	$V_0 \cap V_1 \neq \emptyset$	$\int_{v \in V_0}(a(v) \cdot X) \| \int_{v \in V_1}(b(v) \cdot Y) = \int_{v \in V_0 \cap V_1}((a\|b)(v) \cdot (X \| Y))$
CMI7$'$	$V_0 \cap V_1 = \emptyset$	$\int_{v \in V_0}(a(v) \cdot X) \| \int_{v \in V_1}(b(v) \cdot Y) = \delta(min(sup(V_0), min(V_1)))$
CM8		$(X+Y) \| Z = X \| Z + Y \| Z$
CM9		$X \| (Y+Z) = X \| Y + X \| Z$
D1	$a \notin H$	$\delta_H(a) = a$
D2	$a \in H$	$\delta_H(a) = \delta$
ATD		$\delta_H(a(t)) = \delta_H(a)(t)$
D3		$\delta_H(X+Y) = \delta_H(X) + \delta_H(Y)$
D4		$\delta_H(X \cdot Y) = \delta_H(X) \cdot \delta_H(Y)$

Table 15: ACPρ

Theorem 5.8.1 *Elimination Theorem*
For each term $p \in \mathcal{T}$ *there is a basic term* p_b *such that* BPA$\rho\delta$I $\vdash p = p_b$

Proof. Easy from the combination of the Elimination Theorem 4.2.3 and Theorem 5.4.8 where we prove that every term in BPA$\rho\delta$I has a *basic* term. □

5.9 An Abstract Operational Semantics for ACPρI

We can gather all the action rules from the previous Tables:

Import *atom* rules for $a \neq \delta$ from Table 13 Import rules for $a \neq \delta$ w.r.t. $\cdot, +$ and *(absolute) time shift* from Table 4 Import rules for $a \neq \delta$ w.r.t. $\Vert, \vert, \lfloor\!\lfloor$ and *bounded initialization* from Table 8
$\dfrac{U(p) > L(p)}{p \overset{\delta(U(p))}{\longrightarrow} \surd}$

Table 16: Transition System Specification for ACPρI

Theorem 5.9.1 *The axioms of* ACPρI *are sound with respect of the bisimulation equivalence* $\leftrightarrow$

Proof. Left to the reader □

Theorem 5.9.2 *The operational semantics presented in this section for* ACPρI *is equivalent with the original semantics for* ACPρI*:*
$\forall p, q \in \mathcal{T}^{time-closed}$

$$p \leftrightarrow q \Longleftrightarrow < p, 0 > \leftrightarrow < q, 0 >$$

Proof. Left to the reader □

5.10 An Abstract Operational Semantics for ACPρI in *ntyft* format

In this section we give a Transition System Specification for ACPρI in *ntyft* format. We can take almost all the action rules from the previous Tables. Only for the δ-rules for $\vert$ and $\lfloor\!\lfloor$ we have to give new ones. For example, if we take the following terms with $a\vert\bar{a} \neq \delta$ and $a\vert b = \delta$, we do not expect δ-transition in the first case but we do expect one in the second case.

$$\begin{array}{lll} \int_{v\in<0,1>} a(v) \vert \int_{v\in<0,1>} \bar{a}(v) & \overset{\delta(1)}{\not\longmapsto} & \\ \int_{v\in<0,1>} a(v) \vert \int_{v\in<0,1>} b(v) & \overset{\delta(1)}{\longmapsto} & \surd \end{array}$$

Note that the δ-rule of Table 10 does not generate a δ-transition in any of the two cases.

The point is, that the following implication does not hold anymore:

$$U(p) = u \quad \not\Rightarrow \quad \exists a \in A_\delta \cup A_{\surd}\, \exists p' \in \mathcal{T} \quad p \stackrel{a(u)}{\longmapsto} p'$$

Instead, now we have

$$U(p) = u \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} \exists V\ sup(V) = u\, \forall s \in V \quad \exists a \in A_\delta \cup A_{\surd}\, \exists p' \in \mathcal{T} \quad p \stackrel{a(s)}{\longmapsto} p' \\ \wedge \\ \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad p \stackrel{b(r)}{\not\longmapsto} \end{array} \right\}$$

The obtained δ-rules for $|$ and $\lfloor\!\lfloor$ are given in Table 17. As usual the transition relation is denoted by $\longmapsto$ and the bisimulation equivalence is denoted by $\underline{\leftrightarrow}_{\longmapsto}$. Finally we can "import" all the

$$| \ : \quad \frac{\begin{array}{l} \{ \ \ sup(V) = u\, \forall s \in V \quad \exists a \in A_\delta \cup A_{\surd}\, \exists p' \quad p \stackrel{a(s)}{\longmapsto} p' \\ \quad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad p \stackrel{b(r)}{\not\longmapsto} \\ \quad sup(W) \geq u\, \forall z \in W \quad \exists a \in A_\delta \cup A_{\surd}\, \exists q' \quad q \stackrel{a(z)}{\longmapsto} q' \\ \quad s < u \quad \forall r \geq s\, \forall c, c' \in A_\delta \cup A_{\surd} \quad (p \stackrel{c(u)}{\not\longmapsto} \ \vee q \stackrel{c'(u)}{\not\longmapsto} \ \vee \hat{c}|\hat{c}' = \delta) \ \ \} \end{array}}{p|q \stackrel{\delta(u)}{\longmapsto} \delta\,, \quad q|p \stackrel{\delta(u)}{\longmapsto} \delta}$$

$$\lfloor\!\lfloor \ : \quad \frac{\begin{array}{l} \{ \ \ \exists V\ sup(V) = u\, \forall s \in V \quad \exists a \in A_\delta \cup A_{\surd}\, \exists q' \quad q \stackrel{a(s)}{\longmapsto} q' \\ \quad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad q \stackrel{b(r)}{\not\longmapsto} \\ \quad \exists a' \in A_\delta \cup A_{\surd}\, \exists t > u\, \exists p' \quad p \stackrel{a'(t)}{\longmapsto} p' \qquad \} \end{array}}{p \lfloor\!\lfloor q \stackrel{\delta(u)}{\longmapsto} \delta}$$

$$\frac{\begin{array}{l} \{ \ \ p \stackrel{\delta(u)}{\longmapsto} \delta \\ \quad sup(V) \geq u\, \forall s \in V \quad \exists a \in A_\delta \cup A_{\surd}\, \exists q' \quad q \stackrel{a(s)}{\longmapsto} q' \ \ \} \end{array}}{p \lfloor\!\lfloor q \stackrel{\delta(u)}{\longmapsto} \delta}$$

Table 17: Additional δ-rules for some merge-operators

action rules to obtain an operational semantics for ACPρI. This is given in Table 18

Theorem 5.10.1 *Bisimulation equivalence is a congruence*

Proof. Analogous to 4.4.1 □

Lemma 5.10.2 $\forall p \in \mathcal{T}^{time-closed}\ \forall u \in \mathbb{R}^{\geq 0}$

$$u = U(p) > L(p) \quad \Longleftrightarrow \quad p \stackrel{\delta(u)}{\longmapsto} \delta$$

Import *atom* rules for $a \neq \delta$ from Table 14
Import rules for $a \neq \delta$ w.r.t. $\cdot, +$ and *(absolute) time shift* from Table 6
Import rules w.r.t. $\|, |, \mathbin{⌊\!⌊}$ and *bounded initialization* from Table 9

Table 18: Transition System Specification for ACPρI in *ntyft* format

Proof. See Appendix □

Theorem 5.10.3 *The two Transition System Specifications and the associated definitions of bisimulation induce the same equivalence.*
$\forall p, q \in \mathcal{T}^{time-closed}$

$$p \underline{\leftrightarrow} q \Longleftrightarrow p \underline{\leftrightarrow}_{\longmapsto} q$$

Proof. See the proofs of Theorem 3.4.5 and Theorem 4.4.3 □

6 Concluding Remarks

In this paper we obtained a soundness and completeness result for Real Time Process Algebra with integration. We had to introduce some machinery, such as *basic* terms and *conditional* terms, to come this far. However, it is our belief that these introduced notions have a value in itself. The complexity we encountered was due to the notion of integration, which can be seen as an alternative composition over a continuum associated with a variable binding and scoping mechanism. We restricted ourselves to *prefixed* integration, thus not allowing general integration. However, we don't know any real-life processes which are ruled out by this restriction

We did not consider other work like [Gro90], [MT90], [NS90], [RR88] or [Wan90]. Neither of these papers contain a notion similar to integration and most of the works are based on discrete time. Therefore, these works are less general and less complicated. However, it is an interesting question whether there are common ideas within the field of timed process algebra. For example most authors believe that idling is non-deterministic. In our formalism this is formulated by $t \gg X + t \gg Y = r \gg \{(t-r) \gg X + (t-r) \gg Y\}$ if $r < t$. However, other authors like Groote, and until recently Sifakis as well, believe that idling forces a choice. It would be clarifying to have charted all these similarities and differences.

There are more concepts introduced or touched upon in [BB89] than treated in this paper. Topics for future research are recursion, abstraction and the relation between absolute and relative time. Another interesting question is, whether it is possible to give a finite equational logic and an operational semantics containing only finitely branching transition systems for terms with integration but without unguarded recursion. This would be an important step towards practical applications.

6.1 Acknowledgements

The author wants to thank, first of all, Jos Baeten for his patience in reading all previous versions and for his support and criticism. Thanks go to the author's colleagues Willem Jan Fokkink, Jan

Friso Groote, Henri Korver and Alban Ponse for discussing ideas and for giving help in LaTeX, to Jan Bergstra for his comments on the presentation of the *basic* and *conditional* terms and to Frits Vaandrager for giving a first idea how to define the *basic* terms in the way the are defined now. Finally, Martin Wirsing is thanked for suggesting the research of completeness in ACPρ with integration.

References

[BB89] J.C.M. Baeten and J.A. Bergstra. Real time process algebra. Report P8916, Programming Research Group, University of Amsterdam, 1989. Revised version as report CS-R9053,CWI 1990, to appear in the *Journal of Formal Aspects of Computing* .

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

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

[Bri88] E. Brinksma. *On the design of Extended LOTOS – a specification language for open distributed systems.* PhD thesis, Department of Computer Science, University of Twente, 1988.

[BV89] J.C.M. Baeten and F.W. Vaandrager. An algebra for process creation. Report CS-R8907, CWI, Amsterdam, 1989. Under revision for *Acta Informatica.*

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

[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. Report CS-R8950, CWI, Amsterdam, 1989. An extended abstract appeared 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. Report CS-R9015, CWI, Amsterdam, 1990. An extended abstract appeared in L. Logrippo, R.L. Probert and H. Ural, editors, *Proceedings 10th International Symposium on Protocol Specification, Testing and Verification*, Ottawa, pages 261–274, 1990.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall International, 1985.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science.* Springer-Verlag, 1980.

[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. An earlier version (RT-C16) appeared in M. Broy and C.B. Jones, editors, *Proceedings IFIP Working Conference on Programming Concepts and Methods*, Sea of Gallilea, Israel. North-Holland, 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] R. 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.

A Soundness of the Axioms of BPA$\rho\delta$

A.1 Proof of Theorem 3.2.4

Theorem 3.2.4

The axioms of BPA$\rho\delta$ *are sound w.r.t. to the ultimate delay bisimulation equivalence* $\leftrightarrow_U$.

Proof.

- $A1 - A5$
  Trivial, soundness follows directly from the action rules for $\cdot$ and $+$.

- $ATA1 \quad a(0) = \delta(0)$
  Both $a(0)$ and δ do not have any transitions, moreover $U(a(0)) = U(\delta) = 0$.

- $ATA2 \quad \delta(t) \cdot X = \delta(t)$
  Both $\delta(t) \cdot X$ and $\delta(t)$ do not have any transitions, moreover $U(\delta(t) \cdot X) = U(\delta(t)) = t$.

- $ATA3 \quad \delta(t) + \delta(r) = \delta(r)$ whenever $t \leq r$
  Both $\delta(t) + \delta(r)$ and $\delta(r)$ do not have any transitions, moreover $U(\delta(t) + \delta(r)) = U(\delta(r)) = r$ whenever $t \leq r$.

- $ATA4 \quad a(t) + \delta(t) = a(t)$
  Both terms have only one transition $\xrightarrow{a(t)} \surd$, moreover $U(a(t)+\delta(t)) = max(U(a(t)), U(\delta(t))) = t = U(a(t))$.

- $ATA5 \quad a(t) \cdot X = a(t) \cdot (t \gg X)$
  Note that $U(a(t) \cdot X) = U(a(t)) = t = U(a(t) \cdot (t \gg X))$ and both terms can only perform a transition $\xrightarrow{a(t)}$ resulting in resp. $t \gg X$ and $t \gg (t \gg X)$. Now we have to prove that $t \gg X$ is bisimilar with $t \gg (t \gg X)$. If $t \gg X \xrightarrow{a(r)} X'$ then it must be the case that $r > t$ and this implies $t \gg (t \gg X) \xrightarrow{a(r)} X'$. Moreover, if $U(X) \leq t$ then $U(t \gg X) = t = U(t \gg (t \gg X))$ and otherwise $U(t \gg X) = U(X) = U(t \gg (t \gg X))$. The other direction, for which we have to prove that every transition of $t \gg (t \gg X)$ has a corresponding transition in $t \gg X$ can be done similarly.

- $ATB1 \quad t < r \Longrightarrow t \gg a(r) = a(r)$
  Both terms have only the transition $\xrightarrow{a(r)} \surd$, moreover since $t < r$ we have $U(t \gg a(r)) = r = U(a(r))$.

- ATB2 $t \geq r \Longrightarrow t \gg a(r) = \delta(t)$
$t \gg a(r)$ does not have any transitions because $t \geq r$, of course $\delta(t)$ does not have any transitions either. Bisimulation follows from $U(\delta(t)) = t$ and $U(t \gg a(r)) = t$ whenever $r < t$.

- ATB3 $t \gg (X + Y) = (t \gg X) + (t \gg Y)$
We show that every transition of $(t \gg X) + (t \gg Y)$ has a corresponding transition in $t \gg (X + Y)$. If $(t \gg X + t \gg Y) \xrightarrow{a(r)} X'$ then it must be the case that either $t \gg X \xrightarrow{a(r)} X'$ or $t \gg Y \xrightarrow{a(r)} X'$. Assume it is the first possibility, then it must be the case that $t < r$ and $X \xrightarrow{a(r)} X'$ which implies $t < r$ and $X + Y \xrightarrow{a(r)} X'$ and thus $t \gg (X + Y) \xrightarrow{a(r)} X'$ as well. The other direction can be handled in the same way. Moreover $U(t \gg (X + Y)) = max(t, U(X + Y)) = max(t, max(U(X), U(Y))) = max(max(t, U(X)), max(t, U(Y))) = max(U(t \gg X), U(t \gg Y)) = U((t \gg X) + (t \gg Y))$.

- ATB4 $t \gg (X \cdot Y) = (t \gg X) \cdot Y$
We show that every transition of $t \gg (X \cdot Y)$ has a corresponding transition in $(t \gg X) \cdot Y$. If $t \gg (X \cdot Y)$ has a transition $\xrightarrow{a(r)}$ then there are two cases, either X is atomic and thus $X \equiv a(r)$ or X is not atomic. We look only at the second case. If $t \gg (X \cdot Y) \xrightarrow{a(r)} X' \cdot Y$, then it must be the case that $t < r$ and $X \cdot Y \xrightarrow{a(r)} X' \cdot Y$ which is implied only by $t < r$ and $X \xrightarrow{a(r)} X'$. But then we know that it must also be the case that $t \gg X \xrightarrow{a(r)} X'$ and thus $(t \gg X) \cdot Y \xrightarrow{a(r)} X' \cdot Y$. Moreover $U(t \gg (X \cdot Y)) = max(t, U(X \cdot Y)) = max(t, U(X)) = U(t \gg X) = U((t \gg X) \cdot Y)$

□

B Some Remarks on the Format of the Action Rules

An equivalence relation can model an equational theory only when it is a congruence w.r.t. to all operators. The equivalence on transition systems we are dealing with is the bisimulation equivalence. Following [Gro89] we can obtain bisimulation equivalence as a congruence by using only action rules obeying some conditions. In this section we introduce this elegant result, by discussing a format of action rules such that bisimulation equivalence on the resulting transition systems is automatically a congruence.

In Table 5 we gave an action rule for δ-transitions (δ-rule):

$$\frac{U(p) > L(p)}{p \xrightarrow{\delta(U(p))} \surd}$$

It will be shown in this section that this rule is not in the right format, since the premise $U(p) > L(p)$ depends too much on the syntax of the terms. However we can give a δ-rule for each operator, for example for the $+$:

$$\frac{p \xrightarrow{\delta(r)} p' \quad \forall a \in A, \forall r' \geq r \; q \overset{a(r')}{\not\longrightarrow} \quad \forall s > r \; q \overset{\delta(s)}{\not\longrightarrow}}{p + q \xrightarrow{\delta(r)} p'}$$

which might be read as

$$\frac{p \xrightarrow{\delta(r)} p' \quad L(q) < r \quad U(q) \leq r}{p + q \xrightarrow{\delta(r)} p'}$$

Formally, the premise of an action rule consists of a set of positive premises, denoting the transitions that must hold for applying this action rule, and a set of negative premises, denoting the absence of certain transitions. Thus actually, the given action rule is an abbreviation for

$$\frac{\{p \xrightarrow{\delta(r)} p'\} \cup \{q \overset{b(r')}{\not\longrightarrow} \mid b \in A_\delta\ ,\ r' \geq r\} \cup \{q \overset{\delta(s)}{\not\longrightarrow} \mid s > r\}}{p + q \xrightarrow{\delta(r)} p'}$$

where we make the set of positive premises and the set of negative premises explicit.

Note that although $\delta(1)$ has a $\delta(1)$-transition, $a(2) + \delta(1)$ does not have this $\delta(1)$-transition since it has a $a(2)$-transition.

If we take a closer look at the format of the action rule we recognise *term variables* like p, q and a *constructor* like $+$. Moreover, we have to introduce the definition of $\not\longrightarrow$, the set of "negative" transitions.

Some Definitions w.r.t. action rules an negative premises

- The relation $\not\longrightarrow \subseteq State \times Label$ is induced in a negative way by the relation $\longrightarrow$

$$(\forall q \in State\ :\ (p, \lambda, q) \notin \longrightarrow) \overset{def}{\Longleftrightarrow} (p, \lambda) \in \not\longrightarrow$$

  and we write $p \overset{\lambda}{\not\longrightarrow}$ for $(p, \lambda) \in \not\longrightarrow$. Hence, $p \overset{\lambda}{\not\longrightarrow}$ can be read as "p does not have any λ-labelled transitions".

- $TermVar$ is a set of variables (over $\mathcal{T}$), of which the elements are called *term variables.*

- F is a set of constructors with typical elements f each having a rank $r(f)$. In our case F contains the constructors $+, \cdot$ with $r(+) = r(\cdot) = 2$ and for each $t \in \mathbb{R}^{\geq 0}$ there is a unary constructor $t \gg$ with $r(t \gg) = 1$. Note that we have changed the treatment of the *(absolute) time shift* a little, but in this context we have to deal with single sorted constructors.

- $\mathcal{T}(TermVar, F)$ are the terms constructed by taking variables from $TermVar$ and combining them by constructors from F.

- An action rule of the following form

$$\frac{\{t_k \xrightarrow{a_k(r_k)} y_k | k \in K\} \cup \{t_l \overset{b_l(u_l)}{\not\longrightarrow} | l \in L\}}{f(x_1, .., x_{r(f)}) \xrightarrow{c(s)} t'}$$

  is in *ntyft* format if it obeys the following conditions:

  1. $t_k, t_l, t' \in \mathcal{T}(TermVar, F)$
  2. $y_k, x_i \in TermVar$ with y_k and x_i are pairwise disjoint
  3. $f \in F$
  4. $a_k, b_l, c \in A_\delta$ where A_δ is the alphabet as defined earlier. The variables r_k, u_l and s are called *time variables* and are taken from a set $TVar$.

- An instantiation of an action rule is obtained by substituting terms for the *term variables* and substituting real numbers for *time variables* .

- A *stratification* S is a function which assigns to each transition an element in some totally ordered well founded set, let us say D:

$$n \;:\; State \times Label \times State \longrightarrow D$$

- An action rule of the above form can be *stratified* if for each possible substitution γ

$$\begin{array}{rlcl} & n[\gamma(\,t \xrightarrow{a(r)} y\,)] & \leq & n[\gamma(\,f(x_1,..x_{r(f)}) \xrightarrow{c(s)} t)] \\ \forall l \in L, \forall t_{l,\gamma} & n[\gamma(\,t_l \xrightarrow{b_l(u_l)} t_{l,\gamma}\,)] & < & n[\gamma(\,f(x_1,..x_{r(f)}) \xrightarrow{c(s)} t)] \end{array}$$

- A Transition System Specification is well-founded if every backward chain in its *Variable Dependency Graph* is finite. The set of nodes of this Graph is the set of variables occurring in a positive premise of one of the action rules of the Transition System Specification. There is an arrow from x to y if there is a positive premise $t_k \xrightarrow{a_k(r_k)} y_k$ such that x is a variable occurring in t_k and y is the variable occurring y_k.

Constructing a transition relation from action rules

A transition relation $\mathcal{R}$ is "constructed" from a set of action rules by first initialising $\mathcal{R}^{\,0}$ with all instantiations of the axioms in the set of rules. Next we will apply the inference rules in the set; if there are transitions in $\mathcal{R}^{\,n}$ such that the positive premise of an inference rule can be instantiated and there are no instantiations of the negative premises in $\mathcal{R}^{\,n}$ we obtain the instantiation of the conclusion as a new element of $\mathcal{R}^{\,n+1}$, where $\mathcal{R}^{\,n} \subseteq \mathcal{R}^{\,n+1}$. In this way we can construct the fixed point $\mathcal{R}$.

However we have to be careful, once we used the fact that a negative premise was valid for $\mathcal{R}^{\,i}$ we may not add a transition in $\mathcal{R}^{\,j}$ with $j > i$ neglecting this negative premise. A *stratification* and the well-foundedness guarantees that it will be the case.

Action rules in *ntyft*-format result in a congruence

Now we have defined all the preliminaries concerning action rules with negative premises and their stratifications, we can give the following theorem which is used in the proves of the Lemmas 3.4.2 and 4.4.1.

Theorem B.0.1 *Bisimulation equivalence is a congruence for each well-founded Transition System Specification in which all rules are in ntyft format and in which all the action rules can be stratified.*

Proof. See [Gro89] □

C The Transition System in *ntyft* format has Consistent δ-transitions

C.1 Proof of Lemma 3.4.3 and Lemma 4.4.2

Lemma 3.4.3
$\forall p \in \mathcal{T} \forall u \in \mathbb{R}^{\geq 0}$

$$u = U(p) > L(p) \iff p \overset{\delta(u)}{\longmapsto} \delta$$

Proof. By induction on the *size* of p. We give only some cases, the other cases are left to the reader.

- $p \in A_\delta^{time}$

$$\begin{array}{l} U(p) > L(p) \\ \Longleftrightarrow \\ p \equiv \delta(r) \\ \Longleftrightarrow \\ p \equiv \delta(r) \ \wedge\ \delta(r) \overset{\delta(r)}{\longmapsto} \surd \end{array}$$

- $p \equiv p + q$

$$\begin{array}{llllll}
 & U(p+q) > L(p+q) & & & & \\
\Longleftrightarrow & & & & & \\
\{ & U(p+q) > L(p+q) & \wedge & & \{ & U(p+q) > L(p+q) \ \wedge \\
 & U(p) \geq U(q) & \} & \vee & & U(q) \geq U(p) \quad \} \\
\Longleftrightarrow & & & & & \\
\{ & U(p+q) > L(p) & \wedge & & & \\
 & U(p+q) > L(q) & \wedge & & & \\
 & U(p) \geq U(q) & \} & \vee & .... & \\
\Longleftrightarrow & & & & & \\
\{ & max(U(p), U(q)) > L(p) & \wedge & & & \\
 & max(U(p), U(q)) > L(q) & \wedge & & & \\
 & U(p) \geq U(q) & \} & \vee & .... & \\
\Longleftrightarrow & & & & & \\
\{ & U(p) > L(p) & \wedge & & & \\
 & U(p) > L(q) & \wedge & & & \\
 & U(p) \geq U(q) & \} & \vee & .... & \\
\Longleftrightarrow & & & & & \\
\{ & p \overset{\delta(u)}{\longmapsto} \surd & \wedge & & & \\
 & \forall a \in A \cup A_\surd \ :\ \forall r \geq u \quad p \overset{a(r)}{\not\longmapsto} & \wedge & & & \\
 & \forall b \in A_\delta \cup A_\surd \ :\ \forall r > u \quad q \overset{b(r)}{\not\longmapsto} & \} & \vee & .... & \\
\Longleftrightarrow & & & & & \\
\exists u & u \geq 0 & & & & \\
\{ & \exists p' \ :\ p \overset{\delta(u)}{\longmapsto} p' & \wedge & & & \\
 & \forall a \in A \cup A_\surd \ :\ \forall r \geq u \quad q \overset{a(r)}{\not\longmapsto} & \wedge & & & \\
 & \forall r > u \quad q \overset{\delta(r)}{\not\longmapsto} & \} & \vee \ .... & &
\end{array}$$

- $p \equiv p \| q$

$$\begin{array}{l}
U(p\|q) > L(p\|q) \\
\Longleftrightarrow \\
\left\{ \begin{array}{l} U(p) > L(p\|q) \\ U(p) \leq U(q) \end{array} \right\} \qquad \vee \qquad \left\{ \begin{array}{l} U(q) > L(p\|q) \\ U(q) \leq U(p) \end{array} \right\} \\
\Longleftrightarrow \\
\left\{ \begin{array}{l} U(p) > L(p\|q) \\ U(p) \leq U(q) \\ U(p) = L(p) \end{array} \right\} \vee \left\{ \begin{array}{l} U(p) > L(p\|q) \\ U(p) \leq U(q) \\ U(p) > L(p) \end{array} \right\} \vee \ \ldots \\
\Longleftrightarrow \\
\alpha_1 \qquad\qquad \vee \ \alpha_2 \qquad\qquad \vee \ \ldots
\end{array}$$

We continue with α_1 and α_2 separately.

$$
\begin{array}{ll}
\{\ U(p) > L(p\|q) & \wedge \\
\ \ U(p) \leq U(q) & \wedge \\
\ \ U(p) = L(p) & \}
\end{array}
$$

$$
\begin{array}{lll}
\Longleftrightarrow & & \\
\exists a \in A \cup A_{\surd} & \{\ a(U(p)) \in initact(p) & \wedge \\
& \ \ U(p) \notin inittim(p\|q) & \wedge \\
& \ \ U(p) \leq U(q) & \} \\
\Longleftrightarrow & & \\
\exists a \in A \cup A_{\surd} & \{\ a(U(p)) \in initact(p) & \wedge \\
& \ \ U(p) \notin inittim(p|q) & \wedge \\
& \ \ U(p) \notin inittim(p \mathbin{\lfloor\!\lfloor} q) & \wedge \\
& \ \ U(p) \notin inittim(p \mathbin{\lfloor\!\lfloor} q) & \wedge \\
& \ \ U(p) \leq U(q) & \} \\
\Longleftrightarrow & & \\
\exists a \in A \cup A_{\surd} & \{\ a(U(p)) \in initact(p) & \wedge \\
& \ \ U(p) \notin inittim(p|q) & \wedge \\
& \ \ U(p) \in inittim(p) \Longrightarrow \{U(q) \leq U(p) \Longleftrightarrow U(p) \notin inittim(p \mathbin{\lfloor\!\lfloor} q)\} & \wedge \\
& \ \ U(p) \leq U(q) \Longrightarrow U(p) \notin inittim(q \mathbin{\lfloor\!\lfloor} p) & \wedge \\
& \ \ U(p) \leq U(q) & \} \\
\Longleftrightarrow & & \\
\exists a \in A \cup A_{\surd} & \{\ a(U(p)) \in initact(p) & \wedge \\
& \ \ U(p) = U(q) & \wedge \\
& \ \ U(p) \notin inittim(p|q) & \} \\
\Longleftrightarrow & & \\
\exists a \in A \cup A_{\surd}\ \exists u > 0 & & \\
& \{\ \exists p' \quad p \overset{a(u)}{\longmapsto} p' \qquad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad p \overset{b(r)}{\not\longmapsto} & \wedge \\
& \ \ \exists a' \in A_\delta \cup A_{\surd}\, \exists q' \quad q \overset{a'(u)}{\longmapsto} q' \qquad \forall b \in A_\delta \cup A_{\surd}\, \forall r > u \quad q \overset{b(r)}{\not\longmapsto} & \wedge \\
& \ \ \forall c, c' \in A_\delta \cup A_{\surd} \quad \{p \overset{c(u)}{\not\longmapsto} \vee q \overset{c'(u)}{\not\longmapsto} \vee \hat{c}|\hat{c}' = \delta\} &
\end{array}
$$

And α_2:

$$
\begin{array}{ll}
\{\ U(p) > L(p\|q) & \wedge \\
\ \ U(p) \leq U(q) & \wedge \\
\ \ U(p) > L(p) & \} \\
\Longleftrightarrow & \\
\{\ U(p) > L(p) \quad \Longrightarrow \quad U(p) > L(p \mathbin{\lfloor\!\lfloor} q) & \wedge \\
\ \ U(p) > L(p) \quad \Longrightarrow \quad U(p) > L(p|q) & \wedge \\
\ \ U(p) > L((q \gg U(p)) \mathbin{\lfloor\!\lfloor} p) = L(q \mathbin{\lfloor\!\lfloor} p) & \wedge \\
\ \ U(p) \leq U(q) & \wedge \\
\ \ U(p) > L(p) & \} \\
\Longleftrightarrow & \\
\{\ U(p) \leq U(q) & \wedge \\
\ \ U(p) > L(p) & \} \\
\Longleftrightarrow &
\end{array}
$$

$$
\begin{array}{l}
\exists u > 0 \\
\{ \;\; p \overset{\delta(u)}{\longmapsto} \surd \hspace{12em} \wedge \\
\quad \exists a' \in A_\delta \cup A_\surd \, \exists r \geq u \, \exists q' \quad q \overset{a'(r)}{\longmapsto} q' \;\; \} \\
\Longleftrightarrow \\
\exists u > 0 \\
\{ \;\; \exists p' \quad p \overset{\delta(u)}{\longmapsto} p' \hspace{10em} \wedge \\
\quad \exists a' \in A_\delta \cup A_\surd \, \exists r \geq u \, \exists q' \quad q \overset{a'(r)}{\longmapsto} q' \;\; \}
\end{array}
$$

□

C.2 Proof of Lemma 5.10.2

Lemma 5.10.2
$\forall p \in \mathcal{T} \; \forall u \in \mathbb{R}^{\geq 0}$

$$u = U(p) > L(p) \quad \Longleftrightarrow \quad p \overset{\delta(U(p))}{\longmapsto} \surd$$

Note that $\mathcal{T}$ is now the set of terms concerning *integrals* as defined in section 5.8. We give only the $+$. The rules for $\Vert\!\underline{}$ and $|$ must be changed as well, but their derivations are left to the reader.

Proof. By induction on the *size* of p.

- $p \equiv p + q$

$$
\begin{array}{l}
U(p+q) > L(p+q) \\
\Longleftrightarrow \\
\{ \;\; U(p) > L(p) \hspace{10em} \wedge \\
\quad U(p) > L(q) \hspace{10em} \wedge \\
\quad U(p) \geq U(q) \hspace{10em} \} \;\; \vee \;\; \ldots. \\
\Longleftrightarrow \\
\exists u > 0 \\
\{ \;\; p \overset{\delta(u)}{\longmapsto} \surd \hspace{10em} \wedge \\
\quad u > sup(inittim(q)) \hspace{6em} \wedge \\
\quad u \geq sup(active(q)) \hspace{6em} \} \;\; \vee \;\; \ldots. \\
\Longleftrightarrow \\
\exists u > 0 \\
\{ \;\; \exists p' \, : \, p \overset{\delta(u)}{\longmapsto} p' \hspace{8em} \wedge \\
\quad \exists r < u \, \forall r' \geq r \, \forall b \in A \cup A_\surd \quad q \overset{b(r')}{\not\longmapsto} \quad \wedge \\
\quad \forall b \in A_\delta \cup A_\surd \, \forall t > u \quad q \overset{b(t)}{\not\longmapsto} \hspace{4em} \} \;\; \vee \;\; \ldots \\
\Longleftrightarrow \\
\exists u > 0 \\
\{ \;\; \exists p' \, : \, p \overset{\delta(u)}{\longmapsto} p' \hspace{8em} \wedge \\
\quad \exists r < u \, \forall r' \geq r \, \forall b \in A \cup A_\surd \quad q \overset{b(r')}{\not\longmapsto} \quad \wedge \\
\quad \forall t > u \quad q \overset{\delta(t)}{\not\longmapsto} \hspace{8em} \} \;\; \vee \;\; \ldots
\end{array}
$$

□