1991

J.C.M. Baeten, J.A. Bergstra

Real space process algebra

Computer Science/Department of Software Technology Report CS-R9131 June

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).

Real Space Process Algebra

J.C.M. Baeten
Department of Software Technology, CWI,
P.O.Box 4079, 1009 AB Amsterdam, The Netherlands

Programming Research Group, University of Amsterdam,
P.O.Box 41882, 1009 DB Amsterdam, The Netherlands

J.A. Bergstra
Programming Research Group, University of Amsterdam,
P.O.Box 41882, 1009 DB Amsterdam, The Netherlands

Department of Philosophy, Utrecht University,
Heidelberglaan 2, 3584 CS Utrecht, The Netherlands

We extend the real time process algebra of [BB91] to real space-time process algebra, where actions are not just parametrized by a time coordinate, but also by three spatial coordinates. We describe two versions: classical space-time, where all equations are invariant under Galilei transformations, and relativistic space-time, where all equations are invariant under Lorentz transformations. The latter case in turn splits into two subcases: the temporal interleaving model and the true concurrency model.

1980 Mathematics Subject Classification (1985 revision): 68Q45, 68Q55, 68Q65, 68Q50.
1987 CR Categories: F.4.3, D.2.10, D.3.1, D.3.3.
Key words & Phrases: process algebra, real time, real space-time, Galilei transformation, Lorentz transformation, interleaving, true concurrency.
Note: Partial support received by ESPRIT basic research action 3006, CONCUR, and by RACE contract 1046, SPECS. This document does not necessarily reflect the views of the SPECS consortium.

1. INTRODUCTION.

We quote from PEACKOCK [P30]:

> Algebra may be considered, in its most general form, as *the* science which treats of the combinations of arbitrary signs and symbols by means of defined through arbitrary laws ...

Based on this fairly liberal but certainly classical description of algebra the axiom system ACP (Algebra of Communicating Processes) of BERGSTRA & KLOP [BK84, 85, 86] can be viewed as a part of algebra. In [BB91], the system ACP was extended with real time by having all actions parametrized by some $t \in \mathbb{R}^{\geq 0}$. This constitutes a departure from algebra in the sense that the real numbers are not a purely algebraic concept. As it turns out in the construction of real time process algebra (ACPρ) from ACP, all that matters is that time is organized as a totally ordered set.

In this report we will exploit the fact that ACPρ can be generalized to a setting in which time constitutes a partially ordered set. Rather than working with a partial ordering in general, we concentrate on two examples of such partial orderings that have a particularly useful interpretation.

i. $\langle P^f(\mathbb{R}^3) \times \mathbb{R}, <_c \rangle$, the set of finite nonempty subsets of $\mathbb{R}^3$ with a time coordinate, with $(v,t) <_c (w,r)$ iff $t < r$. We interpret an element (v,t) of $P^f(\mathbb{R}^3) \times \mathbb{R}$ as a set of points in classical space-time, all having the same time coordinate t.

ii. $\langle \mathbb{R}^4, < \rangle$ with $x < y$ if $x \neq y$ and y is in the light cone of x in the sense of special relativity, taking the fourth coordinate as the time component.

Report CS-R9131
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

In this way we obtain real space-time process algebra, which we abbreviate to real space process algebra.

This report deals exclusively with the design of appropriate axiom systems for the description of the various real space process algebras leaving the illustration of its practical use by means of examples as an issue of secondary importance for the moment. Moreover we consider finite processes only. In particular this means that we do not consider recursion and integration.

For motivation and examples of the use of ACP and related systems we refer to [BK84, 85, 86] or [BW90]. Motivation and illustrating examples on real time process algebra can be found in [BB91].

2. REAL TIME PROCESS ALGEBRA

We start with a review of real time process algebra as introduced in [BB91]. We make a few small changes, thereby modifying ACPρ into ACPδρ, in order to facilitate extensions to real space process algebra further on. This does not imply that the theory ACPρ from [BB91] is considered problematic in any way; in fact it may be more useful in 'practice' than the axiom systems to be discussed below.

2.1 BASIC PROCESS ALGEBRA.

Process algebra starts from a given *action alphabet* A (usually finite). Elements a,b,c of A are called *atomic actions*, and are constants of the sort P of *processes*. The theory Basic Process Algebra (BPA) has two binary operators $+,\cdot$: $P \times P \rightarrow P$; + stands for alternative composition and $\cdot$ for sequential composition. BPA has the first five axioms from table 1.

If we add to BPA a special constant δ in P (not in A) standing for *inaction*, comparable to NIL or 0 of CCS (see MILNER [M80, 89] or HENNESSY [HE88]) or STOP of CSP (see HOARE [H85]), we obtain the theory BPAδ. The two axioms for δ are the last two in table 1.

$X + Y = Y + X$	A1
$(X + Y) + Z = X + (Y + Z)$	A2
$X + X = X$	A3
$(X + Y) \cdot Z = X \cdot Z + Y \cdot Z$	A4
$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	A5
$X + \delta = X$	A6
$\delta \cdot X = \delta$	A7

Table 1. BPAδ.

When we add real time to this setting, our basic actions are not from the set $A_\delta = A \cup \{\delta\}$, but from the set

$$AT = \{a(t) \mid a \in A_\delta, t \in \mathbb{R}\} \cup \{\delta\}.$$

The process a(t) performs action a at time t, and then terminates. The process δ(t) deadlocks at time t. The process δ cannot do anything, in particular it cannot wait. Again, these actions can be combined by $+,\cdot$. In this paper we take times t from $\mathbb{R}$ rather than from $\mathbb{R}^{\geq 0} = \{r \in \mathbb{R} \mid r \geq 0\}$. As a consequence, the law $a(0) = \delta(0)$ of [BB91] has to be removed. Instead of the identification $\delta(0) = \delta$ of [BB91], we take a slightly more abstract view and introduce the identification $\delta(t) = \delta$ for all $t \in \mathbb{R}$, thus disregarding the operational difference between various deadlocks. The axiom system ACPρ of [BB91] still has its value and we do not propose to change it. Rather, we will introduce a second theory ACPδρ. This happens with the introduction of BPAδρ. We have:

$$BPA\delta\rho = BPA\rho - \{a(0) = \delta(0)\} + \{\delta(t) = \delta\}.$$

As mentioned above, the law ATA1 of [BB91] (reading $a(0) = \delta$) was changed. The letter A the names of the axioms start with, refers to *absolute time* (versions with relative time were also considered in [BB91], but are not treated here).

As in [BB91], we have the additional operation $\gg$, the *(absolute) time shift.* $t \gg X$ denotes the process X starting at time t. This means that all actions that have to be performed at or before time t are turned into deadlocks because their execution has been delayed too long.

Due to the presence of the law $\delta(t) = \delta$, axioms ATA2,3,4 of [BB91] are derivable from the other axioms of BPAδρ, and so a simplification is possible. In table 2, we have $a \in A_\delta$.

$\delta(t) = \delta$	ATA1*
$a(t) \cdot X = a(t) \cdot (t \gg X)$	ATA5
$t < r \Rightarrow t \gg a(r) = a(r)$	ATB1
$t \geq r \Rightarrow t \gg a(r) = \delta$	ATB2*
$t \gg (X + Y) = (t \gg X) + (t \gg Y)$	ATB3
$t \gg (X \cdot Y) = (t \gg X) \cdot Y$	ATB4

Table 2. BPAδρ.

2.2 Algebra of Communicating Processes.

An axiomatization of parallel composition with communication uses the left merge operator $\mathbin{⌊\!⌊}$, the communication merge operator $|$, and the encapsulation operator ∂_H of [BK84]. Moreover, two extra auxiliary operators introduced in [BB91] are needed: the ultimate delay operator and the bounded initialization operator.

$a \mid b = b \mid a$	C1	$U(a(t)) = \{r \in \mathbb{R} \mid r \geq t\}$	ATU1
$a \mid (b \mid c) = (a \mid b) \mid c$	C2	$U(\delta) = \mathbb{R}$	ATU2
$\delta \mid a = \delta$	C3	$U(X + Y) = U(X) \cap U(Y)$	ATU3
$t \neq r \Rightarrow a(t) \mid b(r) = \delta$	ATC1	$U(X \cdot Y) = U(X)$	ATU4
$a(t) \mid b(t) = (a \mid b)(t)$	ATC2		
		$t \in V \Rightarrow a(t) \gg V = \delta$	ATB5
$X \parallel Y = X \mathbin{⌊\!⌊} Y + Y \mathbin{⌊\!⌊} X + X \mid Y$	CM1	$t \notin V \Rightarrow a(t) \gg V = a(t)$	ATB6
$a(t) \mathbin{⌊\!⌊} X = (a(t) \gg U(X)) \cdot X$	ATCM2	$(X + Y) \gg V = (X \gg V) + (Y \gg V)$	ATB7
$(a(t) \cdot X) \mathbin{⌊\!⌊} Y = (a(t) \gg U(Y)) \cdot (X \parallel Y)$	ATCM3	$(X \cdot Y) \gg V = (X \gg V) \cdot Y$	ATB8
$(X + Y) \mathbin{⌊\!⌊} Z = X \mathbin{⌊\!⌊} Z + Y \mathbin{⌊\!⌊} Z$	CM4		
$(a(t) \cdot X) \mid b(r) = (a(t) \mid b(r)) \cdot X$	CM5'	$\partial_H(a) = a$ if $a \notin H$	D1
$a(t) \mid (b(r) \cdot X) = (a(t) \mid b(r)) \cdot X$	CM6'	$\partial_H(a) = \delta$ if $a \in H$	D2
$(a(t) \cdot X) \mid (b(r) \cdot Y) = (a(t) \mid b(r)) \cdot (X \parallel Y)$	CM7'	$\partial_H(a(t)) = (\partial_H(a))(t)$	ATD
$(X + Y) \mid Z = X \mid Z + Y \mid Z$	CM8	$\partial_H(X + Y) = \partial_H(X) + \partial_H(Y)$	D3
$X \mid (Y + Z) = X \mid Y + X \mid Z$	CM9	$\partial_H(X \cdot Y) = \partial_H(X) \cdot \partial_H(Y)$	D4

Table 3. Remaining axioms of ACPδρ.

The ultimate delay operator U takes a process expression X in CPE, and returns a subset of $\mathbb{R}$. The intended meaning is that $U(X)$ is the set of times at which X must have started, the set of times to which X cannot wait without performing any actions or deadlocking. In [BB91], a slightly different definition is used, and the infimum of the present set is taken, which explains the phrase 'ultimate delay'. In the present approach, we avoid the use of the symbol ω and we prepare for the use of a partially ordered time structure. The bounded initialization operator is also denoted by $\gg$, and is the counterpart of the operator with the same name that we saw in the axiomatization of BPA$\delta\rho$. With $X \gg V$ we denote the process X with its behaviour restricted to the extent that its first action must be performed at a time not in $V \subseteq \mathbb{R}$.

The axioms of ACP$\delta\rho$ are in tables 1 through 3. In table 3, $H \subseteq A$, $a,b,c \in A_\delta$.

A *closed process expression* (CPE) over the signature of ACP$\delta\rho$ with atoms A is an expression that does not contain variables for atoms, processes or real numbers. We allow every real number as a constant, which means there are uncountably many such closed process expressions. For finite closed process expressions an initial algebra can be defined. This is the initial algebra model of ACP$\rho\delta$(A). This structure identifies two closed expressions whenever these can be shown identical by means of application of the axioms. We will look at an operational model next. We denote the set of actions over A without variables by IA (the set of instantiated actions).

2.3 Operational semantics.

We describe an operational semantics for ACP$\delta\rho$ following KLUSENER [K91]. His operational semantics is a simplification of the one in [BB91]. In fact the operational semantics of [K91] is more abstract than the one given in [BB91]. We have two relations

$$\text{step} \subseteq \text{CPE} \times \text{IA} \times \text{CPE} \qquad \text{terminate} \subseteq \text{CPE} \times \text{IA}.$$

The extension of these relations is found as the least fixed point of a simultaneous inductive definition. We write $x \xrightarrow{a(r)} x'$ for step(x, a(r), x'), and $x \xrightarrow{a(r)} \surd$ for terminate(x, a(r)).

Notice that these relations are only defined for $a \in A$, so $a \neq \delta$.

The inductive rules for the operational semantics are similar to those used in structural operational semantics. In table 4, we have $a,b,c \in$ IA, $r,s \in \mathbb{R}$, $x,x',y \in$ CPE.

We see that the action rules for parallel composition make use of the ultimate delay operator. This operator was introduced axiomatically in 2.2, but can also be easily determined for a transition system, since we will always have that $U(x) = \{t \in \mathbb{R} \mid t \geq r \text{ for all } r \text{ with } x \xrightarrow{a(r)} x' \text{ or } x \xrightarrow{a(r)} \surd\}$. Thus, we can avoid the use of the axioms in the description of the structured operational semantics.

2.4 Bisimulations.

Again we consider the class CPE of closed process expressions over ACP$\delta\rho$. A *bisimulation* on CPE is a binary relation R such that

i. for each p and q with R(p, q): if there is a step a(s) possible from p to p', then there is a CPE q' such that R(p', q') and there is a step a(s) possible from q to q'.

ii. for each p and q with R(p, q): if there is a step a(s) possible from q to q', then there is a CPE p' such that R(p', q') and there is a step a(s) possible from p to p'.

iii. for each p and q with R(p, q): a termination step a(s) is possible from p iff it is possible from q.

We say expressions p and q are *bisimilar*, denoted $p \leftrightarrow q$, if there exists a bisimulation on CPE with R(p,q). In [K91] it is shown that bisimulation is a congruence relation on CPE, and that CPE/$\leftrightarrow$ is a model for BPA$\delta\rho$. Indeed, this model is isomorphic to the initial algebra. The advantage of this operational semantics is, that it allows extensions to models containing recursively defined processes.

$$a(r) \xrightarrow{a(r)} \surd$$

$$\frac{x \xrightarrow{a(r)} x'}{x+y \xrightarrow{a(r)} x',\ y+x \xrightarrow{a(r)} x'} \qquad \frac{x \xrightarrow{a(r)} \surd}{x+y \xrightarrow{a(r)} \surd,\ y+x \xrightarrow{a(r)} \surd}$$

$$\frac{x \xrightarrow{a(r)} x'}{x \cdot y \xrightarrow{a(r)} x' \cdot y} \qquad \frac{x \xrightarrow{a(r)} \surd}{x \cdot y \xrightarrow{a(r)} r \gg y}$$

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

$$\frac{x \xrightarrow{a(r)} x',\ r \notin U(y)}{x \| y \xrightarrow{a(r)} x' \| (r \gg y),\ x \llcorner\!\!\llcorner y \xrightarrow{a(r)} x' \| (r \gg y),\ y \| x \xrightarrow{a(r)} (r \gg y) \| x'}$$

$$\frac{x \xrightarrow{a(r)} \surd,\ r \notin U(y)}{x \| y \xrightarrow{a(r)} r \gg y,\ x \llcorner\!\!\llcorner y \xrightarrow{a(r)} r \gg y,\ y \| x \xrightarrow{a(r)} r \gg y}$$

$$\frac{x \xrightarrow{a(r)} x',\ y \xrightarrow{b(r)} y',\ a \mid b = c \neq \delta}{x \| y \xrightarrow{c(r)} x' \| y',\ x \mid y \xrightarrow{c(r)} x' \| y'} \qquad \frac{x \xrightarrow{a(r)} \surd,\ y \xrightarrow{b(r)} \surd,\ a \mid b = c \neq \delta}{x \| y \xrightarrow{c(r)} \surd,\ x \mid y \xrightarrow{c(r)} \surd}$$

$$\frac{x \xrightarrow{a(r)} x',\ y \xrightarrow{b(r)} \surd,\ a \mid b = c \neq \delta}{x \| y \xrightarrow{c(r)} x',\ y \| x \xrightarrow{c(r)} x',\ x \mid y \xrightarrow{c(r)} x',\ y \mid x \xrightarrow{c(r)} x'}$$

$$\frac{x \xrightarrow{a(r)} x',\ r \notin V}{x \gg V \xrightarrow{a(r)} x'} \qquad \frac{x \xrightarrow{a(r)} \surd,\ r \notin V}{x \gg V \xrightarrow{a(r)} \surd}$$

$$\frac{x \xrightarrow{a(r)} x',\ a \notin H}{\partial_H(x) \xrightarrow{a(r)} \partial_H(x')} \qquad \frac{x \xrightarrow{a(r)} \surd,\ a \notin H}{\partial_H(x) \xrightarrow{a(r)} \surd}$$

Table 4. Action rules for ACPδρ.

2.5 GRAPH MODEL.

A graph model (or, more accurately, a *tree* model) of ACPδρ can be constructed as follows.

Process trees are finite directed rooted trees with edges labeled by timed atomic actions and endpoint possibly labeled by δ, satisfying the condition that for each pair of consecutive transitions $s_1 \xrightarrow{a(r)} s_2 \xrightarrow{b(t)} s_3$ it is required that $r < t$.

Now $+, \cdot, \|, \llcorner\!\!\llcorner, \mid, \partial_H, \gg, U, \gg$ can be defined on these graphs in a straightforward manner:

- For +, take the disjoint union of the graphs and identify the roots. If exactly one of the two graphs is the one point δ-graph, remove its label.
- $t \gg g$ is obtained by removing every edge from the root with label $a(r)$ satisfying $t \gg a(r) = \delta$. If this gives a new endpoint, it is labeled by δ.

- $g \cdot h$ is constructed as follows: identify each non-δ endpoint s of g with the root of a copy of $t \gg h$, where t is the time of the edge leading to s.
- $U(g) = \{t \in \mathbb{R} \mid t \geq r \text{ for all } r \in \mathbb{R}, a \in A \text{ and } s \in g \text{ with } root(g) \xrightarrow{a(r)} s\}$.
- Let for $s \in g$ $(g)_s$ denote the subgraph of g with root s. Then $g \| h$ is defined as follows:

- the set of states is the cartesian product of the state sets of g and h, the root the pair of roots.
- transitions: if $s \xrightarrow{a(r)} s'$ and $r \notin U((h)_t)$ then $\langle s,t\rangle \xrightarrow{a(r)} \langle s',t\rangle$;
  if $t \xrightarrow{a(r)} t'$ and $r \notin U((g)_s)$ then $\langle s,t\rangle \xrightarrow{a(r)} \langle s,t'\rangle$;
  if $s \xrightarrow{a(r)} s'$ and $t \xrightarrow{b(r)} t'$ and $a \mid b = c \neq \delta$ then $\langle s,t\rangle \xrightarrow{c(r)} \langle s',t'\rangle$.

Note that the conditions imply that the resulting graph is actually a tree.

- the construction of $g \gg V$, $g \mathbin{\|\!\!\_} h$, $g \mid h$ and $\partial_H(g)$ is now straightforward.

Bisimulation on these graphs is defined as e.g. in BAETEN & WEIJLAND [BW90]. One may prove in a standard fashion that bisimulation is a congruence for all operators of ACPδρ.

Our reason for spelling out the model construction is that it is useful in all other cases arising in this paper as well.

2.10 ZERO OBJECT.

It is interesting to see that we can simplify the system ACPδρ considerably if we take an even more abstract view of the process δ. We do this by adding the additional equation $x \cdot \delta = \delta$. Further on, in the setting of relativistic true concurrency real space process algebra, it seems to be the only option left open to us.

By adding the equation $x \cdot \delta = \delta$, δ is converted into the constant 0 of [BB90]. The axiom system ACP_0' replaces δ by 0. (Notice that ACP_0 denotes the axiom system from [BB90] that combines δ and 0.)

$a \mid b = b \mid a$	C1	$X + Y = Y + X$	A1
$a \mid (b \mid c) = (a \mid b) \mid c$	C2	$(X + Y) + Z = X + (Y + Z)$	A2
$0 \mid a = 0$	CZ3	$X + X = X$	A3
		$(X + Y) \cdot Z = X \cdot Z + Y \cdot Z$	A4
$X \| Y = X \mathbin{\|\!\!\_} Y + Y \mathbin{\|\!\!\_} X + X \mid Y$	CM1	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	A5
$a \mathbin{\|\!\!\_} X = a \cdot X$	CM2	$X + 0 = X$	A6 - Z2
$a \cdot X \mathbin{\|\!\!\_} Y = a \cdot (X \| Y)$	CM3	$0 \cdot X = 0$	A7 - Z3
$(X + Y) \mathbin{\|\!\!\_} Z = X \mathbin{\|\!\!\_} Z + Y \mathbin{\|\!\!\_} Z$	CM4	$X \cdot 0 = 0$	Z1
$a \cdot X \mid b = (a \mid b) \cdot X$	CM5		
$a \mid b \cdot X = (a \mid b) \cdot X$	CM6	$\partial_H(a) = a$ if $a \notin H$	D1
$a \cdot X \mid b \cdot Y = (a \mid b) \cdot (X \| Y)$	CM7	$\partial_H(a) = 0$ if $a \in H$	DZ2
$(X + Y) \mid Z = X \mid Z + Y \mid Z$	CM8	$\partial_H(X + Y) = \partial_H(X) + \partial_H(Y)$	D3
$X \mid (Y + Z) = X \mid Y + X \mid Z$	CM9	$\partial_H(X \cdot Y) = \partial_H(X) \cdot \partial_H(Y)$	D4

Table 5. ACP_0'.

If we now look at ACP0ρ, by again replacing all δ's by 0 in ACPδρ, we can simplify considerably by not using the laws ATCM2,3 but the original laws CM2,3. This can be done because summands that have a wrong time sequence will be removed altogether by 0. Thus we see

$a(1) \| b(2) = a(1) \mathbin{\|\!\!\_} b(2) + b(2) \mathbin{\|\!\!\_} a(1) + a(1) \mid b(2) = a(1) \cdot b(2) + b(2) \cdot a(1) + 0 =$

$= a(1) \cdot b(2) + b(2) \cdot (2 \gg a(1)) = a(1) \cdot b(2) + b(2) \cdot 0 = a(1) \cdot b(2) + 0 = a(1) \cdot b(2)$.

As a consequence, the ultimate delay operator and the bounded initialization operator are not needed anymore for an axiomatization. Thus, ACP0ρ = A1-5 + Z1-3 + ZTA1 + ATA5 + ZTB2 + ATB1,3,4 + C1,2 + CZ3 + ATC1,2 + CM1,4,8,9 + CM2'-7' + D1-4 + ATD. These axioms can be found in tables 2, 3, 5 and 6 below.

$0(t) = 0$	ZTA1
$t \geq r \Rightarrow t \gg a(r) = 0$	ZTB2
$a(t) \mathbin{⫡} X = a(t) \cdot X$	CM2'
$a(t) \cdot X \mathbin{⫡} Y = a(t) \cdot (X \parallel Y)$	CM3'

Table 6. Additional axioms for ACP0ρ.

3. CLASSICAL RSPA.

In Classical Real Space Process Algebra (RSPA-C) we will add space coordinates to all atomic actions. Apart from this, we get a straightforward extension of the theory in section 2. The set of atomic actions is now

$$AST = \{a(x,y,z;t) \mid a \in A_\delta, x,y,z,t \in \mathbb{R}\} \cup \{\delta\}.$$

Instead of $a(x,y,z;t)$, we often write $a(\vec{x};t)$, sometimes also $a(\vec{x})(t)$.

3.1 MULTI-ACTIONS.

Multi-actions are process terms generated by actions from AST and the communication function $\mid$. All multi-actions can be written in a normal form modulo commutativity and associativity. These have the following form:

$$a_1(\vec{x}_1;t) \mid a_2(\vec{x}_2;t) \mid \ldots \mid a_n(\vec{x}_n;t),$$

where $n \geq 1$, all $a_i \neq \delta$ and all $\vec{x}_i$ are pairwise different. For these normal forms, we use the notation

$$(a_1(\vec{x}_1) \,\&\, a_2(\vec{x}_2) \,\&\, \ldots \,\&\, a_n(\vec{x}_n))(t),$$

but we do not introduce & as a function because it would be a partial one. Semantically, in RSPA-C, the multi-actions play the role of the timed actions in RTPA. We use the following axioms to reduce every communication term to normal form. In table 7, we have $a,b,c \in A_\delta$, and u,v,w are multi-actions.

$a \mid b = b \mid a$	C1
$a \mid (b \mid c) = (a \mid b) \mid c$	C2
$\delta \mid a = \delta$	C3
$t \neq r \Rightarrow a(\vec{p};t) \mid b(\vec{q};r) = \delta$	ASC1
$a(\vec{p};t) \mid b(\vec{p};t) = (a \mid b)(\vec{p};t)$	ASC2
$u \mid v = v \mid u$	ASC3
$u \mid (v \mid w) = (u \mid v) \mid w$	ASC4
$\delta \mid u = \delta$	ASC5

Table 7. Communication function for RSPA-C.

3.2 REAL SPACE PROCESS ALGEBRA.

Classical real space process algebra now has exactly the same axioms as real time process algebra, only the letters a,b now do not range over A respectively A_δ, but over expressions of the form $a_1(\vec{x}_1)$ & $a_2(\vec{x}_2)$ & ... & $a_n(\vec{x}_n)$ as above. Note that axiom ATA1* now reads $\delta(\vec{p};t) = \delta$. Further, BPA$\delta\rho\sigma$ contains axioms A1-7, ATA5 and ATB1-4 with a ranging over expressions $a_1(\vec{x}_1)$ & ... & $a_n(\vec{x}_n)$.

The axioms for ultimate delay are again ATU1-4 (with in ATU1 a ranging over the larger set). Parallel composition is dealt with likewise, obtaining the axiom system ACP$\delta\rho\sigma$ by adding axioms CM1,4-9, ATCM2,3, D1-4, ATD. Similarly, we obtain ACP0$\rho\sigma$.

The operational semantics for RSPA-C is just like in the temporal case, in section 2. Transitions are labeled with multi-actions, and these play exactly the same role as the timed actions in the case of ACP$\delta\rho$. Similarly we may define a graph model for ACP$\delta\rho\sigma$. In both cases bisimulation can be defined in the same way.

3.3 LEMMA.

ACP$\delta\rho\sigma$, with the set of atomic actions limited to $\{a(\vec{p};t) \mid a \in A_\delta, t \in R\}$ for a fixed $\vec{p}$, is equivalent to ACP$\delta\rho$.

In words, classical real space process in one point is the same as real time process algebra. The proof of this fact is easy, as all axioms are the same, and with axioms ASC1-2, all multi-actions reduce to single actions.

Relating this axiom system ACP$\delta\rho\sigma$ to the system of [BB91] the following can be said. In [BB91], the system ACPρ has timed δ's and it is extended to incorporate multi-actions at different locations using the equation $\delta(\vec{p};t) = \delta(t)$. Thus, in the setting of ACPρ one can maintain the time of time-space deadlocks in a context with multi-atoms. This mechanism of [BB91] can be called ACP$\rho\sigma$, it has its justification because it preserves more detailed information about the timing of deadlocks.

4. PHYSICS PRELIMINARIES.

We will very briefly mention some notions from physics that will be relevant in the following section. For more information, any text book covering special relativity theory will suffice, see e.g. [B21].

4.1 COORDINATE TRANSFORMATIONS.

First, let us consider a *Galilei transformation.* A Galilei transformation $\mathcal{G}_{\vec{z}}$ over a vector $\vec{z}$ is a translation in the direction of $\vec{z}$ with the length of this vector per unit of time. Thus, if the origins of the two systems coincide, then we get the formula $\mathcal{G}_{\vec{z}}(\vec{p};t) = (\vec{p} - \vec{z}\cdot t;\, t)$.

If the mapping T is projection onto the time coordinate, i.e. $T(\vec{p};t) = t$, then obviously

$$T(\mathcal{G}_{\vec{z}}(\vec{p};t)) = T(\vec{p};t).$$

Thus, in *classical* (Newtonian) mechanics, time is absolute (independent of a linear observer). In special relativity, the same transformation is described by means of a Lorentz transformation.

For the *Lorentz transformation* $\mathcal{L}_{\vec{z}}$, time is not absolute anymore. Thus, it can occur that there are two points in $\mathbb{R}^4$ $\alpha = (\vec{p};t)$ and $\beta = (\vec{q};t)$ (so $T(\alpha) = T(\beta)$), and there are two vectors $\vec{z}$ and $\vec{w}$ in $\mathbb{R}^3$ such that

$$T(\mathcal{L}_{\vec{z}}(\alpha) < T(\mathcal{L}_{\vec{z}}(\beta)) \text{ but } T(\mathcal{L}_{\vec{w}}(\alpha)) > T(\mathcal{L}_{\vec{w}}(\beta)).$$

In fact, for all α,β with $\alpha \neq \beta$ and $T(\alpha) = T(\beta)$ such $\vec{z}$ and $\vec{w}$ can be found. This leads us to consider a new ordering on $\mathbb{R}^4$.

4.2 LIGHT CONES.
We define for $\alpha,\beta \in \mathbb{R}^4$: $\alpha < \beta$ iff for all $\vec{z} \in \mathbb{R}^3$ we have $T(\mathcal{L}_{\vec{z}}(\alpha) < T(\mathcal{L}_{\vec{z}}(\beta))$.
This is a partial ordering on $\mathbb{R}^4$. We write $\alpha \# \beta$ if α and β are incomparable in this ordering. The set $\{\beta \in \mathbb{R}^4 \mid \beta \geq \alpha\}$ is called the *light cone* of α. The intuition behind this is that if β is in the light cone of α, then it is possible to travel from α to β with a speed less than or equal to the speed of light.

4.3 REAL SPACE PROCESS ALGEBRA.
A Galilei transformation can be applied to a closed process expression over RSPA-C using the following inductive definition:

$\mathcal{G}_{\vec{z}}(a(\vec{p};t)) = a(\mathcal{G}_{\vec{z}}(\vec{p};t))$
$\mathcal{G}_{\vec{z}}(x \square y) = \mathcal{G}_{\vec{z}}(x) \square \mathcal{G}_{\vec{z}}(y)$ for $\square = +, \cdot, \|$
$\mathcal{G}_{\vec{z}}(\partial_H(x)) = \partial_H(\mathcal{G}_{\vec{z}}(x))$
$\mathcal{G}_{\vec{z}}(t \gg x) = t \gg (\mathcal{G}_{\vec{z}}(x))$
$\mathcal{G}_{\vec{z}}(x \gg V) = \mathcal{G}_{\vec{z}}(x) \gg V.$

$\mathcal{G}_{\vec{z}}$ acts on a transition system for ACP$\delta\rho\sigma$ by transforming each action occurring in a label. One may easily prove that $g \leftrightarrows h$ implies $\mathcal{G}_{\vec{z}}(g) \leftrightarrows \mathcal{G}_{\vec{z}}(h)$ and that $\mathcal{G}_{\vec{z}}$ commutes with all operators in the graph model.

4.4 AN INCOMPATIBILITY.
Now an important observation is that classical real space process algebra is not compatible with Lorentz transformations. First of all assume that $\mathcal{L}_{\vec{z}}(a(\vec{p};t)) = a(\mathcal{L}_{\vec{z}}(\vec{p};t))$ and $\mathcal{L}_{\vec{z}}(x \parallel y) = \mathcal{L}_{\vec{z}}(x) \parallel \mathcal{L}_{\vec{z}}(y)$. Now we will use the fact that $\mathcal{L}_{\vec{z}}$ does not preserve the partial ordering on $P^f(\mathbb{R}^3) \times \mathbb{R}$, described in the introduction, that underlies ACP$\delta\rho\sigma$.

Let $\alpha \# \beta$ and consider $p = a(\alpha)\cdot b(\beta)$. Suppose without loss of generality that $T(\alpha) < T(\beta)$. Choose $\vec{z}$ such that $T(\mathcal{L}_{\vec{z}}(\alpha)) > T(\mathcal{L}_{\vec{z}}(\beta))$. Write $\alpha' = \mathcal{L}_{\vec{z}}(\alpha)$, $\beta' = \mathcal{L}_{\vec{z}}(\beta)$. Then we find:

$a(\alpha) \parallel b(\beta) = a(\alpha)\cdot b(\beta)$ and $a(\alpha') \parallel b(\beta') = b(\beta')\cdot a(\alpha')$ and thus
$\mathcal{L}_{\vec{z}}(a(\alpha)\cdot b(\beta)) = \mathcal{L}_{\vec{z}}(a(\alpha) \parallel b(\beta)) = a(\alpha') \parallel b(\beta') = b(\beta')\cdot a(\alpha')$,

which contradicts the required distribution over $\cdot$. Hence RSPA-C is not *Lorentz-invariant*, and a different axiomatization is needed for a Lorentz-invariant form of real space process algebra. We will discuss such an axiomatization in the following section.

5. RELATIVISTIC RSPA.

In Relativistic Real Space Process Algebra (RSPA-R) we consider actions parametrized by elements of $\mathbb{R}^4$. The set of atomic actions is now

$$AST = \{a(x,y,z,t) \mid a \in A_\delta, x,y,z,t \in \mathbb{R}\} \cup \{\delta\}.$$

Instead of $a(x,y,z,t)$, we often write $a(\alpha)$.

5.1 BASIC RELATIVISTIC REAL SPACE PROCESS ALGEBRA.

Here, we consider BPAδρσ in a relativistic setting. Since time and space cannot be separated anymore, we will replace an expression of the form $t \gg X$ by an expression $\alpha \gg X$ for $\alpha \in \mathbb{R}^4$. Similarly, we obtain BPA0ρσr, by changing δ into 0 in axioms STA1, STB2. The operational semantics for BPAδρσr is as expected, replacing labels a(t) by labels a(α). Also, bisimulation is defined similarly.

$\delta(\alpha) = \delta$	STA1
$a(\alpha) \cdot X = a(\alpha) \cdot (\alpha \gg X)$	STA5
$\alpha < \beta \;\Rightarrow\; \alpha \gg a(\beta) = a(\beta)$	STB1
$\neg(\alpha < \beta) \;\Rightarrow\; \alpha \gg a(\beta) = \delta$	STB2
$\alpha \gg (X + Y) = (\alpha \gg X) + (\alpha \gg Y)$	STB3
$\alpha \gg (X \cdot Y) = (\alpha \gg X) \cdot Y$	STB4

Table 8. BPAδρσr = BPAδ + STA1,5 + STB1-4.

5.2 ULTIMATE DELAY.

The definition of the ultimate delay operator now uses the light cone of a point in four-space.

$U(a(\alpha)) = \{\beta \in \mathbb{R}^4 \mid \beta \geq \alpha\}$	STU1
$U(\delta) = \mathbb{R}^4$	STU2
$U(X + Y) = U(X) \cap U(Y)$	STU3
$U(X \cdot Y) = U(X)$	STU4

Table 9. Ultimate delay operator.

5.3 PARALLEL COMPOSITION.

First, we consider what happens when we take exactly the same axioms for parallel composition as before. First, the bounded initialization operator is defined as before, but now has signature $P \times Pow(\mathbb{R}^4) \to P$.

Then, we look at axioms for the communication function. We cannot consider multi-actions of simultaneously happening actions at different locations anymore, because simultaneity is not Lorentz-invariant: simultaneity of two actions is only due to the 'lucky' choice of an inertial system. This explains the replacement of axiom ASC1 by the following axiom STC1:

$\alpha \neq \beta \;\Rightarrow\; a(\alpha) \mid b(\beta) = \delta$ STC1

$a(\alpha) \mid b(\alpha) = (a \mid b)(\alpha)$ STC2.

Then, the axiom system ACPδρσr is obtained by taking BPAδρσr + STU1-4 + STB5-8 + C1-3 + STC1,2 and adding the axioms CM1,4-9, ATCM2,3, D1-4, ATD as in 2.2, but replacing all symbols t,r by α,β.

The operational rules for parallel composition are as before. Process graphs are just as in the RSPA-C case, be it that now for a pair of consecutive transitions $s \xrightarrow{a(\alpha)} s' \xrightarrow{b(\beta)} s''$ it is needed that $\alpha < \beta$.

5.4 LORENTZ INVARIANCE.

We can define a Lorentz transformation of a process in the obvious way:

$\mathcal{L}_{\vec{z}}(a(\alpha)) = a(\mathcal{L}_{\vec{z}}(\alpha))$

$\mathcal{L}_{\vec{z}}(x \,\square\, y) = \mathcal{L}_{\vec{z}}(x) \,\square\, \mathcal{L}_{\vec{z}}(y)$ for $\square = +, \cdot, \|$

$\mathcal{L}_{\vec{Z}}(\partial_H(x)) = \partial_H(\mathcal{L}_{\vec{Z}}(x))$
$\mathcal{L}_{\vec{Z}}(\alpha \gg x) = \mathcal{L}_{\vec{Z}}(\alpha) \gg \mathcal{L}_{\vec{Z}}(x)$
$\mathcal{L}_{\vec{Z}}(x \gg V) = \mathcal{L}_{\vec{Z}}(x) \gg \mathcal{L}_{\vec{Z}}(V)$.

We claim that with this definition, all closed process identities provable from ACPδρσ are still provable after a Lorentz transformation of both sides of the equality sign. Thus, we can say that ACPδρσ is Lorentz invariant.

Notice that a transition system obtained from a finite closed process expression using the transition rules above is a finite graph with actions as edge labels and such that along every path through the graph, each action is in the light cone of the previous action.

It is also straightforward to define a Lorentz transformation of a transition system: just transform all action labels. We find that bisimulation is Lorentz invariant, and that modulo bisimulation, all operators commute with $\mathcal{L}_{\vec{Z}}$.

5.5 TEMPORAL INTERLEAVING.

Now let us consider an example, to see how these axioms work out. Suppose $\alpha \# \beta$. Then we find:

$$\begin{aligned}
a(\alpha) \| b(\beta) &= a(\alpha) \lfloor\!\lfloor b(\beta) + b(\beta) \lfloor\!\lfloor a(\alpha) + a(\alpha) \mid b(\beta) = \\
&= (a(\alpha) \gg U(b(\beta)) \cdot b(\beta) + (b(\beta) \gg U(a(\alpha)) \cdot a(\alpha) + \delta = \\
&= (a(\alpha) \gg \{\gamma \mid \gamma \geq \beta\}) \cdot b(\beta) + (b(\beta) \gg \{\gamma \mid \gamma \geq \beta\}) \cdot a(\alpha) = \\
&= a(\alpha) \cdot b(\beta) + b(\beta) \cdot a(\alpha) = \\
&= a(\alpha) \cdot (\alpha \gg b(\beta)) + b(\beta) \cdot (\beta \gg a(\alpha)) = \\
&= a(\alpha) \cdot \delta + b(\beta) \cdot \delta.
\end{aligned}$$

We see that ACPδρσ describes a *mono-processor* execution of a parallel composition: there is a single processor executing $X \| Y$, that travels (maybe with the speed of light) from one action to the next. This explains the δ's in the expression above. We see that this semantics for parallel composition is opposed to a so-called 'true concurrency' interpretation, where actions at different locations can happen independently. This example also shows that the system ACP0ρσ has unwanted behaviour: if $\alpha \# \beta$ then the expression $a(\alpha) \| b(\beta)$ will equal 0. Therefore, we will not consider a system with temporal interleaving and the constant 0.

It should be noticed that the equation $a(\alpha) \| b(\beta) = a(\alpha) \cdot b(\beta) + b(\beta) \cdot a(\alpha)$ in fact excludes a parallel execution of $a(\alpha)$ and $b(\beta)$ if $\alpha \# \beta$. Therefore a more appropriate semantics cannot satisfy the expansion theorem. Thus, we will consider a true concurrency semantics, giving up the expansion theorem.

5.6 MULTIPLE PROCESSORS, TRUE CONCURRENCY.

As said above, the axiom system ACPδρσ is sound for a description of single processor execution of concurrent processes. We call this the *temporal interleaving* semantics for real space process algebra. Moreover, the equations of ACPδρσ are Lorentz invariant with respect to the outlined bisimulation model. In this case the system $a(\alpha) \| b(\beta)$ for $\alpha \# \beta$ equals $a(\alpha) \cdot \delta + b(\beta) \cdot \delta$. It follows that $a(\alpha)$ and $b(\beta)$ exclude one another. Assuming that different actions can be performed by different processors it is not at all the case that both actions in $a(\alpha) \| b(\beta)$ exclude one another (if $\alpha \# \beta$). Thus, a multi-processor (or true concurrency) interpretation of merge will be different from the temporal interleaving one. In fact the equation $a(\alpha) \| b(\beta) = a(\alpha) \cdot b(\beta) + b(\beta) \cdot a(\alpha)$ will *not* hold for $\alpha \# \beta$. To put it another way, a Lorentz invariant formulation of multiple processor execution of concurrent systems negates the expansion theorem, and thereby leads to 'true concurrency'.

Looking for a Lorentz invariant formulation of truly concurrent merge in the setting of ACP, we found that ACP itself contains an obstacle that has to be removed. Although we cannot prove this rigorously, it seems that ACP as such has a classical (non-relativistic) bias built in. This bias has to do with the constant δ and can be removed by changing δ into 0. Thus, we will only consider this case with the constant 0.

5.7 SYNTAX.

To recapitulate, we have the following syntax:

P	the sort of processes
A	a finite set of action labels
0	constant; $A_0 = A \cup \{0\}$
$a(\alpha)$	for every $\alpha \in \mathbb{R}^4$ and $a \in A_0$, $a(\alpha) \in P$ is an atomic action
$\mid$	communication function: $A_0 \times A_0 \to A_0$, commutative, associative, 0 as zero element
$+, \cdot, \Vert$	binary operators $P \times P \to P$
$\gg$	binary operator $\mathbb{R}^4 \times P \to P$
$\gg$	binary operator $P \times Pow(\mathbb{R}^4) \to P$
∂_H	unary operators $P \to P$ (for each $H \subseteq A$).

Now we will first consider a truly concurrent operational semantics for this syntax.

5.8 DEFINITION.

A *multiple processor transition system* is a rooted, edge-labeled finite graph such that the edge labels are actions in $A(\mathbb{R}^4)$ and such that for every path the sequence of edge labels $a_0(\alpha_0)\ a_1(\alpha_1) \dots a_n(\alpha_n)$ is such that for all $i < j$ it is not the case that $\alpha_j \leq \alpha_i$. In words: no action in the graph causally precedes an earlier action. Moreover, there is an additional requirement: if there are two paths in the graph that have the same starting point and the same end point, then these paths have the same multi-set of edge labels. Note that we do not restricts ourselves to *trees* any more. We call the trivial one-point graph the *zero graph.*

The Lorentz transformation of a multiple processor transition system is obtained by transforming all edge labels. Since a Lorentz transformation does not change the ordering on $\mathbb{R}^4$, the image is again a multiple processor transition system. Bisimulation is defined as usual.

5.9 PROPOSITION.

If $g \leftrightarrow h$ then $\mathcal{L}_{\vec{z}}(g) \leftrightarrow \mathcal{L}_{\vec{z}}(h)$.

5.10 ULTIMATE DELAY.

Let g be a multiple processor transition system. For a node s of g we denote the subgraph with root s by $(g)_s$. Now let s be a node of g, then $U((g)_s)$ is defined as follows:

- if $(g)_s$ is the zero graph then $U((g)_s) = \mathbb{R}^4$;
- otherwise, suppose s has outgoing edges labeled $a_i(\alpha_i)$ to nodes s_i $(1 \leq i \leq k)$. Then
  $U((g)_s) = \bigcap_{1 \leq i \leq k} (\{\beta \in \mathbb{R}^4 \mid \beta \geq \alpha_i\} \cup U((g)_{s_i}))$.

Next, we can define the operators on multiple processor transition systems.

5.11 DEFINITION.

Let g,h be two multiple processor transition systems.

- graph g+h is obtained by identifying the roots of g and h; this new node is the root.
- graph g·h is obtained as follows: if g or h is the zero graph, then g·h is the zero graph; otherwise, append a copy of h at each endpoint of g (identifying the root of h with the endpoint of g); then remove each branch in h containing an edge with action label b(β) for which $\alpha_1 \gg (\alpha_2 \gg \ldots (\alpha_n \gg b(\beta)\ldots)) = 0$, where $\{\alpha_1, \ldots, \alpha_n\}$ is the multi-set of locations appearing in a path leading to this endpoint.
- graph $\partial_H(g)$ is obtained by removing every branch containing an edge with a label a(α) with $a \in H$.
- graph g∥h is the cartesian product of graphs g and h, with:

  there is an edge $(s,t) \xrightarrow{a(\alpha)} (s^*,t)$ in g∥h if there is an edge $s \xrightarrow{a(\alpha)} s^*$ in g and $\alpha \notin U((h)_t)$

  there is an edge $(s,t) \xrightarrow{b(\alpha)} (s,t^*)$ in g∥h if there is an edge $t \xrightarrow{b(\alpha)} t^*$ in h and $\alpha \notin U((g)_s)$

  there is an edge $(s,t) \xrightarrow{c(\alpha)} (s^*,t^*)$ in g∥h if there are edges $s \xrightarrow{a(\alpha)} s^*$ in g and $t \xrightarrow{b(\alpha)} t^*$ in h with $a \mid b = c \neq 0$.
- graph $\alpha \gg g$ is obtained by removing every branch containing an edge with label a(β) with $\alpha \gg a(\beta) = 0$.

We can check that in each case, the resulting graph is again a multiple processor transition system.

5.12 PROPOSITION.

Bisimulation is a congruence on the set of multiple processor transition systems with operators defined as in 5.11.

5.13 AN AXIOMATIZATION.

We can give an axiomatization for the multiple processor theory, called ACP0ρσrm, with the use of the auxiliary operators $a(\alpha) \circ$ (for each $a \in A$ and $\alpha \in \mathbb{R}^4$), ⋙. The first is a kind of *prefix multiplication*, a unary operator for each atomic action. Whereas the 'closed' multiplication sign · denotes a causal dependency, the 'open' multiplication ∘ does not: we have that actions following ∘ *may not causally precede* the action in front of the ∘ sign. In addition to the time shift operator ≫, we have the *weak time shift operator* ⋙. $\alpha \ggg X$ is a variant of $\alpha \gg X$, that differs in case points in four-space are incomparable. We also use operators ⫫, | , but here they do not imply causal dependencies (but rather causal independencies). Since we have a system with 0, we do not need the ultimate delay operator or the bounded initialization operator.

5.14 REMARK.

Although for α # β we now have $a(\alpha) \| b(\beta) = a(\alpha) \circ b(\beta) + b(\beta) \circ a(\alpha)$, one cannot say that ACP0ρσrm implies arbitrary interleaving, because ∘ is not ·. ∘ does not correspond to sequential composition of systems, rather $a(\alpha) \circ b(\beta)$ is $a(\alpha) \| b(\beta)$ under the additional constraint that b(β) does not causally preceed a(α). A philosophical difficulty arises as follows: consider $\alpha,\beta \in \mathbb{R}^4$ with α # β. Then $a(\alpha) \circ b(\beta)$ and $b(\beta) \circ a(\alpha)$ show the same behaviour for any observer. Still, they are not identified. It follows that ∘ is a truly hidden function. The additional structure imposed by this operator is not always based on an observable (behavioural) criterion. Rather, ∘ is to be seen (as ⫫ and | in this case) as an auxiliary operator that mainly has a formal role to play.

$X + Y = Y + X$	A1	$a \mid b = b \mid a$	C1
$(X + Y) + Z = X + (Y + Z)$	A2	$a \mid (b \mid c) = (a \mid b) \mid c$	C2
$X + X = X$	A3	$0 \mid a = 0$	CZ3
$(X + Y) \cdot Z = X \cdot Z + Y \cdot Z$	A4		
$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	A5	$\alpha \neq \beta \Rightarrow a(\alpha) \mid b(\beta) = 0$	STC1
$X + 0 = X$	A6 - Z2	$a(\alpha) \mid b(\alpha) = (a \mid b)(\alpha)$	STC2
$0 \cdot X = 0$	A7 - Z3		
$X \cdot 0 = 0$	Z1	$X \parallel Y = X \mathbin{|\!\!\lfloor} Y + Y \mathbin{|\!\!\lfloor} X + X \mid Y$	CM1
$a(\alpha) \circ 0 = 0$	OZ1	$a(\alpha) \mathbin{|\!\!\lfloor} X = a(\alpha) \circ X$	CM2'
$0 \circ X = 0$	OZ2	$a(\alpha) \circ X \mathbin{|\!\!\lfloor} Y = a(\alpha) \circ (X \parallel Y)$	CM3'
$0(\alpha) = 0$	MTA1	$(X + Y) \mathbin{|\!\!\lfloor} Z = X \mathbin{|\!\!\lfloor} Z + Y \mathbin{|\!\!\lfloor} Z$	CM4
$a(\alpha) \circ X = a(\alpha) \circ (\alpha \ggg X)$	MTA5	$a(\alpha) \circ X \mid b(\beta) = (a(\alpha) \mid b(\beta)) \circ X$	MTCM5
$a(\alpha) \cdot X = a(\alpha) \circ (\alpha \gg X)$	OA1	$a(\alpha) \mid b(\beta) \circ X = (a(\alpha) \mid b(\beta)) \circ X$	MTCM6
$(a(\alpha) \circ X) \cdot Y = a(\alpha) \circ (X \cdot (\alpha \gg Y))$	OA2	$a(\alpha) \circ X \mid b(\beta) \circ Y = (a(\alpha) \mid b(\beta)) \circ (X \parallel Y)$	MTCM7
$\alpha < \beta \Rightarrow \alpha \gg a(\beta) = a(\beta)$	STB1	$(X + Y) \mid Z = X \mid Z + Y \mid Z$	CM8
$\alpha \# \beta \Rightarrow \alpha \gg a(\beta) = 0$	STB1/2	$X \mid (Y + Z) = X \mid Y + X \mid Z$	CM9
$\alpha \geq \beta \Rightarrow \alpha \gg a(\beta) = 0$	STB2		
$\alpha \gg (X + Y) = (\alpha \gg X) + (\alpha \gg Y)$	STB3		
$\alpha \gg (X \cdot Y) = (\alpha \gg X) \cdot Y$	STB4	$\partial_H(a) = a \quad \text{if } a \notin H$	D1
$\alpha \gg (a(\beta) \circ X) = (\alpha \gg a(\beta)) \circ (\alpha \gg X)$	OTB1	$\partial_H(a) = 0 \quad \text{if } a \in H$	DZ2
$\alpha < \beta \Rightarrow \alpha \ggg a(\beta) = a(\beta)$	MTB1	$\partial_H(a(\alpha)) = \partial_H(a)(\alpha)$	STD
$\alpha \# \beta \Rightarrow \alpha \ggg a(\beta) = a(\beta)$	MTB1/2	$\partial_H(X + Y) = \partial_H(X) + \partial_H(Y)$	D3
$\alpha \geq \beta \Rightarrow \alpha \ggg a(\beta) = 0$	MTB2	$\partial_H(X \cdot Y) = \partial_H(X) \cdot \partial_H(Y)$	D4
$\alpha \ggg (X + Y) = (\alpha \ggg X) + (\alpha \ggg Y)$	MTB3	$\partial_H(a(\alpha) \circ X) = \partial_H(a(\alpha)) \circ \partial_H(X)$	D5
$\alpha \ggg (X \cdot Y) = (\alpha \ggg X) \cdot Y$	MTB4		
$\alpha \ggg (a(\beta) \circ X) = (\alpha \ggg a(\beta)) \circ (\alpha \ggg X)$	OTB2		

Table 10. ACP0pσrm.

We also remark that the notion of a state is not self evident in this theory. Given the transition $a(\alpha) \parallel b(\beta) \xrightarrow{a(\alpha)} b(\beta)$ one might say that $b(\beta)$ is a state of the process $a(\alpha) \parallel b(\beta)$. But if $\alpha \# \beta$ then for some observer, β precedes α in time. This observer cannot possibly observe the state $b(\beta)$. Thus, states are formal mathematical entities rather than 'real' observable quantities.

5.15 NOTE.

Notice that the law $a(\alpha) \cdot X = a(\alpha) \circ (\alpha \gg X)$ (OA1) of the table above implies the law $a(\alpha) \cdot X = a(\alpha) \cdot (\alpha \gg X)$, for all closed terms. For, it is straightforward to derive $\alpha \gg X = \alpha \gg (\alpha \gg X)$ for all closed terms by structural induction, and using this, we obtain:

$$a(\alpha) \cdot X = a(\alpha) \circ (\alpha \gg X) = a(\alpha) \circ (\alpha \gg (\alpha \gg X)) = a(\alpha) \cdot (\alpha \gg X).$$

5.16 NOT EQUAL TO ZERO.

We can add a predicate $\neq 0$ as in [BB90]. Axioms for this unary predicate on processes are in table 11.

$a \neq 0 \Rightarrow a(\alpha) \neq 0$	MNZ1
$\alpha \gg x \neq 0 \Rightarrow a(\alpha) \cdot x \neq 0$	MNZ2
$\alpha \ggg x \neq 0 \Rightarrow a(\alpha) \circ x \neq 0$	MNZ3
$x \neq 0 \Rightarrow x + y \neq 0$	MNZ4

Table 11. Nonzero predicate.

5.17 ACTION RULES.
We can give action rules for ACP0pσrm, but not in a modular fashion as done for systems above. The reason for this is, that the steps that a process e.g. x·y can take, are not completely determined by the steps that x can take, but also on the fact whether y will evaluate to 0 starting at a point where x terminates. Thus, action rules can only defined *on top of* the axioms, as in [BB90].

5.18 CONCLUSION.
We have obtained an important *non-interleaving* model of the ACP syntax. An expansion theorem for this model has a very different interpretation. We conclude that the ACP axiomatization of ‖ using ⌊⌊ and | is still possible in the case of relativistic multi-processor systems, and a truly concurrent Lorentz invariant operational semantics can be found.

Of course many problems are left open, e.g. can one find a *complete* axiomatization of the theory of closed finite process identities in the multi-processor bisimulation model outlined above.

REFERENCES.

[BB91] J.C.M. BAETEN & J.A. BERGSTRA, *Real time process algebra,* Formal Aspects of Computing 3 (2), 1991, pp. 142-188 (original version: report P8916, Programming Research Group, University of Amsterdam 1989).

[BB90] J.C.M. BAETEN & J.A. BERGSTRA, *Process algebra with a zero object,* in: Proc. CONCUR 90, Amsterdam (J.C.M. Baeten & J.W. Klop, eds.), Springer LNCS 458, 1990, pp. 83-98.

[BW90] J.C.M. BAETEN & W.P. WEIJLAND, *Process algebra,* Cambridge Tracts in TCS 18, Cambridge University Press 1990.

[BK84] J.A. BERGSTRA & J.W. KLOP, *Process algebra for synchronous communication,* Inf. & Control 60, 1984, pp. 109-137.

[BK85] J.A. BERGSTRA & J.W. KLOP, *Algebra of communicating processes with abstraction,* TCS 37, 1985, pp. 77-121.

[BK86] J.A. BERGSTRA & J.W. KLOP, *Process algebra: specification and verification in bisimulation semantics,* in: Math. & Comp. Sci. II (M. Hazewinkel, J.K. Lenstra & L.G.L.T. Meertens, eds.), CWI Monograph 4, North-Holland, Amsterdam 1986, pp. 61-94.

[B21] M. BORN, *Die Relativitätstheorie Einsteins,* Springer Verlag 1921.

[HE88] M. HENNESSY, *Algebraic theory of processes,* MIT Press 1988.

[H85] C.A.R. HOARE, *Sequential communicating processes,* Prentice Hall, 1985.

[K91] A.S. KLUSENER, *Completeness in real time process algebra,* report CS-R9106, CWI Amsterdam 1991. Extended abstract in this volume.

[M80] R. MILNER, *A calculus of communicating systems,* Springer LNCS 92, 1980.

[M89] R. MILNER, *Communication and concurrency,* Prentice Hall, 1989.

[P30] G. PEACKOCK, *A treatise of algebra,* Cambridge 1830.