Discrete Time Process Algebra

J.C.M. Baeten
Department of Computing Science, Eindhoven University of Technology,
P.O.Box 513, 5600 MB Eindhoven, The Netherlands

J.A. Bergstra
Programming Research Group, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
and
Department of Philosophy, Utrecht University,
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

The axiom system ACP of [BEK84a] was extended with real time features in [BAB91]. Here we proceed to define a discrete time extension of ACP, along the lines of ATP [NIS94]. We present versions based on relative timing and on absolute timing. Both approaches are integrated using parametric timing. The time free ACP theory is embedded in the discrete time theory.
Note: Partial support received from ESPRIT Basic Research Action 7166, CONCUR2.

1. INTRODUCTION.

Process algebra in the form of ACP [BEK84a, BAW90], CCS [MIL89] or CSP [BRHR84] describes the main features of imperative concurrent programming without explicit mention of time. Implicitly, time is present in the interpretation of sequential composition: in $p \cdot q$ (ACP notation) the process p must be executed before q. A quantative view on the relation between process execution and progress of time is absent in these calculi, however. Process algebras can be developed that provide standardised features to incorporate a quantative view on time. An option is to represent time by means of non-negative reals, and to have time stamps on actions. This is done in [BAB91] for ACP, in [MOT90] for CCS and in [RER88] for CSP. The timed versions of ACP, CCS and CSP in these papers differ concerning the degree to which time stamping is explicit in the notational format of the proposed process algebras. This in turn influences the form of equations, axiomatisations and the appearance of examples.

Another option is to divide time in slices indexed by natural numbers, to have an implicit or explicit time stamping mechanism that provides each action with the time slice in which it occurs and to have a time order within each slice only. This has been worked out in ATP [NSVR90], [NIS94], a process algebra that adds time slicing to a version of ACP based on action prefixing rather than sequential composition. Further, [GRO90a] has extended ACP with time slices whereas [MOT89] have added these features to CCS. We propose to use the phrase *discrete time process algebra* if an enumeration of time slices is used.

The objective of this paper is to extend ACP to a discrete time process algebra. We present three views on discrete time process algebra. In section 3, we consider discrete time process algebra with *relative timing*, where timing refers to the execution of the previous action. We refer to [BAB92] for a

version where all actions have a time stamp. Here, we present the so-called two-phase version, where the passage of time and the execution of actions is separated. In section 4, we have discrete time process algebra with *absolute timing*, where all timing refers to an absolute clock. Here again, we only consider the two-phase version. In section 5, we have discrete time process algebra with *parametric timing*, where absolute and relative timing are integrated. For parametric timing, we introduce a model based on time spectrum sequences.

There are many practical uses conceivable for timed process algebras. In particular, we mention the TOOLBUS (see [BEK94, 95]). This TOOLBUS contains a program notation called T which is syntactically sugared discrete time process algebra. Programs in T are called T-scripts. The runtime system is also described in terms of discrete time process algebra. By using randomised symbolic execution the TOOLBUS implementation enacts that the axioms of process algebra can be viewed as correctness preserving transformations of T-scripts. A comparable part of disrete time process algebra that is used to describe T-scripts has also been used for the description of ϕSDL, flat SDL, a subset of SDL that leaves out modularisation and concentrates on timing aspects (see [BEM95]).

We design our algebras (or rather their specifications) in a modular, incremental way. We have absolute time algebras BPA_{dat}, PA_{dat}, ACP_{dat}, relative time algebras BPA_{drt}, PA_{drt}, ACP_{drt}, and parametric time algebras BPA_{dpt}, PA_{dpt}, ACP_{dpt}. This segmentation resembles the segmentation of ACP that can be found in [BAW90]. ACP_{dat} can be seen as the discretised version of the real time theory ACPρ of [BAB91], ACP_{dpt} the discretised version of the real time theory ACPρ√ of [BAB93].

It appears that ACP_{drt} is an axiom system quite similar to ATP of [NIS94]. The key new operator is a new process constructor $\sigma_{rel}: P \to P$. The notation σ has been taken from [HER90], the subscript rel draws attention to the relative time setting. $\sigma_{rel}(x)$ delays the process x till the next time slice. We can define all operators of ATP [NSVR90, NIS94] in ACP_{drt}. These features include: unit delay $\lfloor x \rfloor(y)$ and maximal progress composition ⊕. It turns out that these operators can be eliminated in the presence of those of ACP_{drt}. (We notice that ATP's ⊕ is already present as + in TCCS of [MOT89].) As ACP_{drt}, ATP and TCCS each use strong bisimulation equivalence to obtain a semantic domain, they may be considered as different (and intertranslatable) sets of generators for the same semantic world of processes. The view of [NIS94] that progress of time by itself should not be allowed to introduce non-determinism has had major impact on our work. If time is represented by an action σ then $\sigma\cdot(x + y) = \sigma\cdot x + \sigma\cdot y$ is an appropriate axiom called time factorisation in [NIS94] (the proposals of [GRO90a], [GM91] do not satisfy time factorisation).

ACKNOWLEDGEMENT: We thank members of the PAM seminar, in particular Michel Reniers (Eindhoven University of Technology), for their helpful remarks.

2. ACP WITH IMMEDIATE TIME STOP.

As the basis for our development, we start out from the time free theory ACP, with an additional constant $\dot{\delta}$. This process will play a special role, it stands for immediate (and catastrophic) deadlock. The syntax has constants a (for each $a \in A$, some given set of atomic actions), a constant δ (inaction) and a constant $\dot{\delta}$ (immediate deadlock). We have operators + (alternative composition) and · (sequential

composition. These elements constitute the syntax of $BPA_{\delta}^{\bullet}$. The syntax of $PA_{\delta}^{\bullet}$ has in addition the parallel composition $\parallel$ and the left-merge $\between$. We also need the absolute value operator $|_|$ in the formulation of the axioms for left merge, as the axioms without this operator will not hold in the extension to relative time. The absolute value operator initialises a process in the first time slice. In a time free theory as is the case here, it is simply the identity. In case of ambiguities or overlap with additional syntax one may write $|_|_{abs}$ for the absolute value operator. The syntax of $ACP_{\delta}^{\bullet}$ adds to this the communication merge $\mid$ (where this operator is given on $A \cup \{\delta\}$, satisfying axioms C1-3) and the encapsulation operator ∂_H (for $H \subseteq A$).

The theory $BPA_{\delta}^{\bullet}$ has the axioms A1-5, A6A, A7, A6ID, A7ID from table 1 below, the theory $PA_{\delta}^{\bullet}$ has in addition the axioms AV1-5, LMID1,2, CM2AV, CM3AV, CM4 and the axiom M1, which is CM1 without third summand ($X \parallel Y = X \between Y + Y \between X$), the theory $ACP_{\delta}^{\bullet}$ has all axioms from table 1. We see that the absolute value operator is the identity for all closed terms over this syntax, and distributes through all operators. The theory ACP of [BEK84a] can be obtained from the theory $ACP_{\delta}^{\bullet}$ as a subalgebra of reduced model specification (an SRM specification, in the terminology of [BAB94]): consider the initial algebra of the theory $ACP_{\delta}^{\bullet}$, reduce this model by omitting the constant $\overset{\bullet}{\delta}$ and the absolute value operator, then the resulting subalgebra is completely axiomatised by the theory ACP. This means that the present axiomatisation induces the same identities between closed terms over ACP.

$X + Y = Y + X$		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	$\delta \mid a = \delta$		C3
$(X + Y)\cdot Z = X\cdot Z + Y\cdot Z$		A4	$X \parallel Y = X \between Y + Y \between X + X \mid Y$		CM1
$(X\cdot Y)\cdot Z = X\cdot(Y\cdot Z)$		A5	$\overset{\bullet}{\delta} \between X = \overset{\bullet}{\delta}$		LMID1
$a + \delta = a$		A6A	$X \between \overset{\bullet}{\delta} = \overset{\bullet}{\delta}$		LMID2
$\delta\cdot X = \delta$		A7	$a \between (\|X\| + \delta) = a\cdot(\|X\| + \delta)$		CM2AV
$X + \overset{\bullet}{\delta} = X$		A6ID	$a\cdot X \between (\|Y\| + \delta) = a\cdot(X \parallel (\|Y\| + \delta))$		CM3AV
$\overset{\bullet}{\delta}\cdot X = \overset{\bullet}{\delta}$		A7ID	$(X + Y) \between Z = X \between Z + Y \between Z$		CM4
			$\overset{\bullet}{\delta} \mid X = \overset{\bullet}{\delta}$		CMID1
$\|\overset{\bullet}{\delta}\| = \overset{\bullet}{\delta}$		AV1	$X \mid \overset{\bullet}{\delta} = \overset{\bullet}{\delta}$		CMID2
$\|a\| = a$		AV2	$a\cdot X \mid b = (a \mid b)\cdot X$		CM5
$\|X + Y\| = \|X\| + \|Y\|$		AV3	$a \mid b\cdot X = (a \mid b)\cdot X$		CM6
$\|X\cdot Y\| = \|X\|\cdot Y$		AV4	$a\cdot X \mid b\cdot Y = (a \mid b)\cdot(X \parallel Y)$		CM7
$\|X \between Y\| = \|X\| \between \|Y\|$		AV5	$(X + Y) \mid Z = X \mid Z + Y \mid Z$		CM8
$\|X \mid Y\| = \|X\| \mid \|Y\|$		AV6	$X \mid (Y + Z) = X \mid Y + X \mid Z$		CM9
$\|\partial_H(X)\| = \partial_H(\|X\|)$		AV7			
			$\partial_H(\overset{\bullet}{\delta}) = \overset{\bullet}{\delta}$		DID
$\partial_H(a) = a$	if $a \notin H$	D1	$\partial_H(X + Y) = \partial_H(X) + \partial_H(Y)$		D3
$\partial_H(a) = \delta$	if $a \in H$	D2	$\partial_H(X\cdot Y) = \partial_H(X)\cdot\partial_H(Y)$		D4

TABLE 1. $ACP_{\delta}^{\bullet}$.

2.2 STRUCTURED OPERATIONAL SEMANTICS.

We give a semantics in terms of Plotkin-style operational rules. As set of states S we have the set of closed process expressions, and we define the following relations on states:

- action step $\subseteq S \times A \times S$, notation $s \xrightarrow{a} s'$ (denotes action execution)
- action termination $\subseteq S \times A$, notation $s \xrightarrow{a} \surd$ (execution of a terminating action)
- immediate deadlock $\subseteq S$, notation $ID(s)$ (holds only for terms equal to $\delta^{\bullet}$).

A relation holds on S just when it derivable (as defined in [VER94]) from the following rules. We omit rules for the absolute value operator, as they are trivial.

$$ID(\delta^{\bullet}) \qquad a \xrightarrow{a} \surd$$

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

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

$$\frac{x \xrightarrow{a} x',\ \neg\ ID(y)}{x \| y \xrightarrow{a} x' \| y,\ y \| x \xrightarrow{a} y \| x',\ x \between y \xrightarrow{a} x' \| y}$$

$$\frac{x \xrightarrow{a} \surd,\ \neg\ ID(y)}{x \| y \xrightarrow{a} y,\ y \| x \xrightarrow{a} y,\ x \between y \xrightarrow{a} y} \qquad \frac{ID(x)}{ID(x \| y),\ ID(y \| x),\ ID(x \between y),\ ID(y \between x)}$$

$$\frac{x \xrightarrow{a} x',\ y \xrightarrow{b} y',\ a \mid b = c}{x \| y \xrightarrow{c} x' \| y',\ x \mid y \xrightarrow{c} x' \| y'} \qquad \frac{x \xrightarrow{a} x',\ y \xrightarrow{b} \surd,\ a \mid b = c}{x \| y \xrightarrow{c} x',\ x \mid y \xrightarrow{c} x',\ y \| x \xrightarrow{c} x',\ y \mid x \xrightarrow{c} x'}$$

$$\frac{x \xrightarrow{a} \surd,\ y \xrightarrow{b} \surd,\ a \mid b = c}{x \| y \xrightarrow{c} \surd,\ x \mid y \xrightarrow{c} \surd} \qquad \frac{ID(x)}{ID(x \mid y),\ ID(y \mid x)}$$

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

TABLE 2. Operational rules for $ACP_{\delta}^{\bullet}$.

In these rules, we find the negation of a predicate. In order to show that these rules give rise to a unique transition relation on closed terms, it helps to use results and terminology of [VER94]. Bisimulation is defined as usual, so a symmetric binary relation R on process terms is a bisimulation iff the following transfer conditions hold:

i. if $R(p,q)$ and $p \xrightarrow{a} p'$ $(a \in A)$, then there is q' such that $q \xrightarrow{a} q'$ and $R(s',t')$;
ii. if $R(p,q)$, then $p \xrightarrow{a} \surd$ $(a \in A)$ iff $q \xrightarrow{a} \surd$.
iii. if $R(p,q)$, then $ID(p)$ iff $ID(q)$.

Two terms p,q are bisimilar, $p \leftrightarrows q$, if there exists a bisimulation relating them. We can make the set of process terms modulo bisimulation into a model for $ACP_\delta^\bullet$. In order to do this, we first need to know that bisimulation is a congruence. [VER94] proves that this is true, as long as the operational rules satisfy his *panth* format. It is easy to establish that this is indeed the case. We can also prove that the axiomatisation given in table 1 is complete for this model.

3. DISCRETE TIME PROCESS ALGEBRA WITH RELATIVE TIMING.

We present axioms for process algebras in which actions are timed relatively. With $cts(a)$ we denote the process that will execute a in the time slice in which it is initialised, and will then terminate immediately. So if $cts(a)$ is enabled during slice 7, then a will be performed in the course of slice 7. With the operator σ_{rel} processes can be delayed one time slice. So if the process $\sigma_{rel}(\sigma_{rel}(cts(a) + cts(b))$ is initialised in slice 5, this has the effect that in slice 7 a choice between a and b must be made. The algebra ACP_{drt} presented is a version of ATP tailored towards ACP. It is a slight modification of the axiom system ACP_{dt} introduced in [BAB92].

3.1 BASIC PROCESS ALGEBRA.

The signature of BPA_{drt}^- has besides the constant $\dot{\delta}$, constants $cts(a)$ (for $a \in A_\delta$), denoting a in the current time slice, with immediate termination. The superscript $^-$ denotes that time free atoms are not part of the signature. We can also denote $cts(a)$ as $\underline{a}[1]$ or $\underline{\underline{a}}$ (the alternative notations were used in [BAB92]). Within a time slice, there is no explicit mention of the passage of time, we can see the passage to the next time slice as a clock tick. Thus, the $cts(a)$ are the nondelayable actions as can be found e.g. in [NIS94]: the action must occur before the next clock tick. The operators are alternative and sequential composition, and the relative discrete time unit delay σ_{rel} (adapted from [HER90], can also be denoted $\sigma_{rel(1)}$ or σ_d; the last notation was used in [BAB92]). The process $\sigma_{rel}(x)$ will start x after one clock tick, i.e. in the next time slice.

The signature of BPA_{drt} adds to this the constants a, for $a \in A_\delta$. These are the delayable actions, a number of clock ticks can occur before the action is executed. More explicitly, we can refer to these constants as $ats(a)$, action a performed in any time slice, with immediate termination. In the presence of an empty process or a silent step, we will have different versions, that do not show immediate termination. The axioms of BPA_{drt}^- are A1-5, A6ID, A7ID of $BPA_\delta^\bullet$ plus DRT1-4 of table 3 below, BPA_{drt} adds to this A6A, A7 and the axiom ADRT.

The axiom DRT1 is the time factorization axiom: it says that the passage of time by itself cannot determine a choice. Axiom DRT3 identifies processes that can be distinguished: $cts(\delta)$ denotes a deadlock at the end of the current time slice, and $\sigma_{rel}(\dot{\delta})$ denotes an immediate deadlock at the beginning of the next time slice. This is the difference between open and closed time stops that we discussed in

[BAB93], section 8.1. Here, as in [BAB93], we abstract from this difference. This simplifies the algebra considerably, but makes the operational semantics somewhat more complicated.

$\sigma_{rel}(X) + \sigma_{rel}(Y) = \sigma_{rel}(X + Y)$	DRT1	$a = cts(a) + \sigma_{rel}(a)$	ADRT
$\sigma_{rel}(X)\cdot Y = \sigma_{rel}(X\cdot Y)$	DRT2		
$\sigma_{rel}(\delta^{\bullet}) = cts(\delta)$	DRT3	$cts(a) + cts(\delta) = cts(a)$	DRT4

TABLE 3. $BPA_{drt} = BPA^{\bullet}_{\delta} + DRT1\text{-}4, ADRT$.

3.2 STRUCTURED OPERATIONAL SEMANTICS.

We give a semantics in terms of Plotkin-style operational rules, by adding the following relation on states: time step $\subseteq S \times S$, notation $s \xrightarrow{\sigma} s'$ (denotes passage to the next time slice).

We add to the rules given in section 2 for $BPA^{\bullet}_{\delta}$ the following rules.

$$cts(a) \xrightarrow{a} \surd \qquad \frac{\neg\Box ID(x)}{\sigma_{rel}(x)\Box \xrightarrow{\sigma} \Box x} \qquad a \xrightarrow{\sigma} a \qquad \delta \xrightarrow{\sigma} \delta$$

$$\frac{x\Box \xrightarrow{\sigma} \Box x'}{x\cdot y\Box \xrightarrow{\sigma} \Box x'\cdot y} \qquad \frac{x\Box \xrightarrow{\sigma} \Box x', \Box y \not\xrightarrow{\sigma}}{\Box x+y\Box \xrightarrow{\sigma} \Box x'+y'} \qquad \frac{x\Box \xrightarrow{\sigma} \Box x', \Box y \not\xrightarrow{\sigma}}{x+y\Box \xrightarrow{\sigma} \Box x', \Box y+x\Box \xrightarrow{\sigma} \Box x'}$$

TABLE 4. Additional operational rules for BPA_{drt}.

This time, we also have the negation of a relation. $y \not\xrightarrow{\sigma}$ means that there is no expression y' such that $y \xrightarrow{\sigma} y'$. Nevertheless, we can still use the results of [VER94]. Bisimulation will also have to preserve time steps, so for a bisimulation R we require in addition:

iv. if $R(p,q)$ and $p \xrightarrow{\sigma} p'$, then there is q' such that $q \xrightarrow{\sigma} q'$ and $R(s',t')$;.

3.3 BASIC TERMS.

We define the set of basic terms B_{drt} over BPA_{drt} inductively. We use the auxiliary sets B^{1}_{drt} (basic terms that start in the current time slice) and B^{∞}_{drt} (basic terms that can start after arbitrary many ticks), to be defined simultaneously.

1. $B^{1}_{drt} \subseteq B_{drt}$
2. $B^{\infty}_{drt} \subseteq B_{drt}$
3. if $a \in A$, then $cts(a) \in B^{1}_{drt}$
4. $\delta \in B^{\infty}_{drt}$
5. if $a \in A$ and $t \in B_{drt}$, then $cts(a)\cdot t \in B^{1}_{drt}$
6. if $a \in A$, then $a \in B^{\infty}_{drt}$
7. if $t,s \in B^{1}_{drt}$, then $t+s \in B^{1}_{drt}$
8. if $a \in A$ and $t \in B_{drt}$, then $a\cdot t \in B^{\infty}_{drt}$
9. $\delta^{\bullet} \in B_{drt}$
10. if $t,s \in B^{\infty}_{drt}$, then $t+s \in B^{\infty}_{drt}$
11. if $t \in B_{drt}$, then $\sigma_{rel}(t) \in B_{drt}$
12. if $t \in B^{1}_{drt}$ and $s \in B_{drt}$, then $t + \sigma_{rel}(s) \in B_{drt}$.

Basic terms over BPA_{drt} are defined by clauses 1-12; we obtain basic terms over BPA^{-}_{drt} by omitting clauses 2,4,6,8,10. Some reflection leads to the insight that the basic terms over BPA^{-}_{drt} are exactly the terms of one of the following three forms:

a. $\dot{\delta}$

b. $\sum_{i<n} \Box cts(a_i)\cdot t_i + \sum_{j<m} \Box cts(b_j)$, with $n+m>0$, $a_i, b_j \in A$ and t_i basic;

c. $\sum_{i<n} \Box cts(a_i)\cdot t_i + \sum_{j<m} \Box cts(b_j) + \sigma_{re}(t)$, with $n,m\geq 0$, $a_i, b_j \in A$ and t,t_i basic.

In the case of BPA_{drt}, we can have extra summands of the form a ($a \in A_\delta$) or $a\cdot t$ in form b.

3.4 ELIMINATION.

Let t be a closed $BPA_{\overline{drt}}$-term. Then there is a basic term s over $BPA_{\overline{drt}}$ such that $BPA_{\overline{drt}} \vdash t = s$.
Let t be a closed BPA_{drt}-term. Then there is a basic term s over BPA_{drt} such that $BPA_{drt} \vdash t = s$.

3.5 SOUNDNESS AND COMPLETENESS.

Let p,q be closed $BPA_{\overline{drt}}$-expressions. Then we have $BPA_{\overline{drt}} \vdash p=q \Leftrightarrow p \leftrightarrows q$,
i.e. $BPA_{\overline{drt}}$ is sound and complete for the set of transition graphs modulo bisimulation.
Let p,q be closed BPA_{drt}-expressions. Then we have $BPA_{drt} \vdash p=q \Leftrightarrow p \leftrightarrows q$,
so also BPA_{drt} is a sound and complete axiomatisation.

3.6 CONSERVATIVITY.

BPA_{drt} is a conservative extension of $BPA^{\bullet}_{\delta}$, i.e. for all closed $BPA^{\bullet}_{\delta}$-terms s,t we have

$$BPA^{\bullet}_{\delta} \vdash s=t \Leftrightarrow BPA_{drt} \vdash s=t.$$

3.7 GRAPH MODEL.

Thus, we have found a sound and complete model for $BPA_{\overline{drt}}$. It is also possible to define a graph model for BPA_{drt} directly, not by giving a transition system specification, but by defining an interpretation of the constants and operators on process graphs. We define a set of process graphs as in [BAW90] with labels from $A \cup \{\sigma\}$ satisfying the extra condition that every node has *at most one* outgoing σ-labeled edge. Moreover, a termination node can have a label $\surd$ (for succesful termination), or ID (for immediate deadlock). A σ-edge may not lead to a $\surd$-node.

Let $\mathcal{G}$ be the set of such process graphs. To state this precisely, an element of $\mathcal{G}$ is a sixtuple $\langle N, E, r, T, ID, \surd\rangle$ where N is the set of *nodes*, $E \subseteq (N\text{-}T \times A\cup\{\sigma\} \times N$ is the set of *edges*, $r \in N$ is the *root node*, $T \subseteq N$ is the set of *termination nodes*, and $ID \subseteq T$, $\surd \subseteq T\text{-}ID$.

We define an interpretation as follows ($n \in \mathbb{N}$):

1. The constant $\dot{\delta}$ is mapped to the graph with 1 node, the root, labeled ID.
2. The constant $cts(a)$ is mapped to the process graph with 2 nodes, the first the root, the last a $\surd$-node, connected by an a-edge. The constant a adds to this graph a σ-edge from the root to itself.
3. The constant $cts(\delta)$ is mapped to the process graph with 1 node, the root, and no edges, no node-labels. The constant δ adds to this graph a σ-edge from the root to itself.
4. Given graphs g,h, we first have to root-unwind both graphs in order to form the sum (see e.g. [BAW90]). In case one graph is the $\dot{\delta}$-graph, the sum is simply the other graph. Then, the graph $g+h$ is

formed by identifying the roots of g and h. Next, if both roots have an outgoing σ-edge, both these edges are removed, and a new σ-edge is added to the sum of the graphs the original σ-edges where going to. If necessary, the procedure is repeated.

5. Given graph g,h, in order to form $g \cdot h$, append at each $\surd$-node of g a copy of h.

6. Given graph g, the tree $\sigma_{rel}(g)$ is formed by removing the ID-label in case g is the $\dot{\delta}$-graph, and otherwise, by adding a new root, with a σ-edge from the new root to the old root.

3.8 PROJECTION.

We can also define a model using a projective limit construction as in [BAB84b]. In order to do this, we define projections. The relative time nth projection operator π^{rel}_n cuts off a process after n steps have been executed. Here, we count both time steps and action steps. We give axioms in table 5 below. We have that every closed term over the extended signature, including projection operators, can also be written as a basic term.

To give an example, $\pi^{rel}_0(cts(a)\cdot cts(\delta)) = \dot{\delta}$, $\pi^{rel}_1(cts(a)\cdot cts(\delta)) = cts(a)\cdot\dot{\delta}$, and $\pi^{rel}_n(cts(a)\cdot cts(\delta)) = cts(a)\cdot cts(\delta)$ for $n \geq 2$. The projective limit model has as domain sequences of closed terms $(p_n)_{n \in \mathbb{N}}$ such that $\pi^{rel}_n(p_{n+1}) = p_n$. Operators are defined pointwise. The given axiomatisation for BPA_{drt} is also sound and complete for this model.

$\pi^{rel}_0(X) = \dot{\delta}$	PRR1	$\pi^{rel}_{n+1}(cts(a)) = cts(a)$	PRR4
$\pi^{rel}_n(\dot{\delta}) = \dot{\delta}$	PRR2	$\pi^{rel}_{n+1}(cts(a)\cdot X) = cts(a)\cdot\pi^{rel}_n(X)$	PRR5
$\pi^{rel}_{n+1}(\sigma_{rel}(X)) = \sigma_{rel}(\pi^{rel}_n(X))$	PRR3	$\pi^{rel}_n(X + Y) = \pi^{rel}_n(X) + \pi^{rel}_n(Y)$	PRR6

TABLE 5. Projections.

Next, we introduce parallel composition. We start by describing a system with merge without communication, a so-called free merge. Different from the real time case of [BAB91], in the discrete time case, it is possible to consider merge without communication, as we will have $cts(a) \parallel cts(b) = cts(a)\cdot cts(b) + cts(b)\cdot cts(a)$.

3.9 FREE MERGE.

In PA_{drt}, we add the parallel composition operator $\parallel$ (merge) with its auxiliary operator $\lfloor\!\lfloor$ (left merge). PA^{-}_{drt} has the axioms of BPA^{-}_{drt}, axioms M1, CM4, LMID1,2 of section 2 and the axioms in table 6 below ($a \in A$, $x,y,z \in P$). Notice that $\sigma_{rel}(X) \lfloor\!\lfloor cts(\delta) = cts(\delta)$ can be derived using DRT7, DRT3, LMID2. PA_{drt} adds the additional axioms of BPA_{drt} and the axioms CM2AV, CM3AV of section 2. We postpone the definition of the absolute value operator on relative time processes to section 5.1. We can still use axioms CM2AV, CM3AV to eliminate the merge from time free terms, in other cases we can use axiom ADRT.

The operational semantics adds the rule in table 7 to the rules concerning $PA^{\bullet}_{\delta}$ in table 2 and the rules of BPA_{drt}.

$cts(a) \mathbin{\mathbb{L}} (X + cts(\delta)) = cts(a)\cdot(X + cts(\delta))$	DRTCM2
$cts(a)\cdot X \mathbin{\mathbb{L}} (Y + cts(\delta)) = cts(a)\cdot(X \parallel (Y + cts(\delta)))$	DRTCM3
$\sigma_{rel}(X) \mathbin{\mathbb{L}} (cts(a) + Y) = \sigma_{rel}(X) \mathbin{\mathbb{L}} (cts(\delta) + Y)$	DRT5
$\sigma_{rel}(X) \mathbin{\mathbb{L}} (cts(a)\cdot Y + Z) = \sigma_{rel}(X) \mathbin{\mathbb{L}} (cts(\delta) + Z)$	DRT6
$\sigma_{rel}(X) \mathbin{\mathbb{L}} \sigma_{rel}(Y) = \sigma_{rel}(X \mathbin{\mathbb{L}} Y)$	DRT7

TABLE 6. $PA_{drt} = BPA_{drt}$ + M1, CM4, LMID1,2, DRTCM2,3, CM2AV, CM3AV, DRT5-7.

$$\frac{x \xrightarrow{\sigma} x',\ y \xrightarrow{\sigma} y'}{x \parallel y \xrightarrow{\sigma} x' \parallel y',\ x \mathbin{\mathbb{L}} y \xrightarrow{\sigma} x' \mathbin{\mathbb{L}} y'}$$

TABLE 7. Additional operational rule for PA_{drt}.

We have results as before:
Let t be a closed PA_{drt}^{-}-term. Then there is a basic term s over BPA_{drt}^{-} such that $PA_{drt}^{-} \vdash t = s$.
Let t be a closed PA_{drt}-term. Then there is a basic term s over BPA_{drt} such that $PA_{drt} \vdash t = s$.
Let p,q be closed PA_{drt}^{-}-expressions. Then we have $PA_{drt}^{-} \vdash p=q \Leftrightarrow p \leftrightarrows q$,
Let p,q be closed PA_{drt}-expressions. Then we have $PA_{drt} \vdash p=q \Leftrightarrow p \leftrightarrows q$.
PA_{drt} is a conservative extension of $PA_{\delta}^{\bullet}$.
As an example, we calculate $(cts(a) + \sigma_{rel}(cts(b))) \parallel (cts(b) + \sigma_{rel}(cts(a))) =$
$= cts(a)\cdot(cts(b) + \sigma_{rel}(cts(a))) + cts(b)\cdot(cts(a) + \sigma_{rel}(cts(b))) + \sigma_{rel}(cts(a)\cdot cts(b) + cts(b)\cdot cts(a))$.

3.10 COMMUNICATION.

We add the communication merge $|$ based on a given commutative, associative function on A_{δ} with δ as neutral element, and the encapsulation operator ∂_H. ACP_{drt}^{-} modifies PA_{drt}^{-} by replacing axiom M1 by CM1 and adding axioms C1-3, CMID1,2, CM8,9, D3,4 of $ACP_{\delta}^{\bullet}$ and the additional axioms in table 8, ACP_{drt} adds also the remaining axioms of $ACP_{\delta}^{\bullet}$ to PA_{drt}. Operational rules are given in table 9. We obtain elimination, soundness, completeness, conservativity as before.

$cts(a) \mid cts(b) = cts(a \mid b)$	DRTC	$cts(a) \mid \sigma_{rel}(X) = cts(\delta)$	DRT8
$cts(a)\cdot X \mid cts(b) = cts(a \mid b)\cdot X$	DRTCM5	$\sigma_{rel}(X) \mid cts(a) = cts(\delta)$	DRT9
$cts(a) \mid cts(b)\cdot X = cts(a \mid b)\cdot X$	DRTCM6	$cts(a)\cdot X \mid \sigma_{rel}(Y) = cts(\delta)$	DRT10
$cts(a)\cdot X \mid cts(b)\cdot Y = cts(a \mid b)\cdot(X \parallel Y)$	DRTCM7	$\sigma_{rel}(X) \mid cts(a)\cdot Y = cts(\delta)$	DRT11
$\partial_H(cts(a)) = cts(a)$ if $a \notin H$	DRTD1	$\sigma_{rel}(X) \mid \sigma_{rel}(Y) = \sigma_{rel}(X \mid Y)$	DRT12
$\partial_H(cts(a)) = cts(\delta)$ if $a \in H$	DRTD2	$\partial_H(\sigma_{rel}(X)) = \sigma_{rel}(\partial_H(X))$	DRT13

TABLE 8. $ACP_{drt} = PA_{drt}$ - M1 + $ACP_{\delta}^{\bullet}$ + DRTC, DRTCM5-7, DRTD1,2, DRT8-13.

$$\frac{x \xrightarrow{\sigma} x',\ y \xrightarrow{\sigma} y'}{x \mid y \xrightarrow{\sigma} x' \mid y'} \qquad \frac{x \xrightarrow{\sigma} x'}{\partial_H(x) \xrightarrow{\sigma} \partial_H(x')}$$

TABLE 9. Additional operational rules of ACP_{drt}.

An example of the elimination of the merge is as follows (with $a \mid b = c$):

$(cts(a) + \sigma_{rel}(cts(b)) \between (cts(b) + \sigma_{rel}(cts(a))) = cts(a)\cdot(cts(b) + \sigma_{rel}(cts(a))) +$

$+ cts(b)\cdot(cts(a) + \sigma_{rel}(cts(b))) + cts(c) + \sigma_{rel}(cts(a)\cdot cts(b) + cts(b)\cdot cts(a) + cts(c))$.

4. DISCRETE TIME PROCESS ALGEBRA WITH ABSOLUTE TIMING.

We present a version of the theory in section 3 using absolute timing, where all timing is related to a global clock. Similar to section 3, we do not consider a version with timestamped actions (as we did in [BAB92]), but instead separate the execution of actions from the passage of time, a *two-phase* theory in the terminology of [NIS94].

4.1 BASIC PROCESS ALGEBRA.

We start with constants $fts(a)$, denoting a in the first time slice ($a \in A_\delta$), followed by immediate termination. An alternative notation is $\underline{a}(1)$. Besides, we have the constant $\dot{\delta}$, and operators $+,\cdot$ as before. In addition, we have the absolute discrete time unit delay σ_{abs}. Axiom DAT7 uses the absolute value operator. This operator is the identity for all processes using absolute timing only. The axioms of BPA_{dat}^- are A1-5, A6ID, A7ID of $BPA_\delta^\bullet$, AV1,3,4 of $PA_\delta^\bullet$, plus DAT1-7, AV8,9 of table 10 below, BPA_{dat} adds to this A6A, A7, AV2 and the axiom ADAT. Notice that we can derive $X + fts(\delta) = X$ for all closed terms X except $\dot{\delta}$, i.e. for all closed terms X with $\neg\, ID(X)$. Also notice that $fts(\delta)\cdot X = fts(\delta)$ can be derived for all closed terms.

$\sigma_{abs}(\dot{\delta}) = fts(\delta)$	DAT1	$\lvert fts(a)\rvert = fts(a)$	AV8
$fts(a) + fts(\delta) = fts(a)$	DAT2	$\lvert\sigma_{abs}(X)\rvert = \sigma_{abs}(\lvert X\rvert)$	AV9
$\sigma_{abs}(X) + \sigma_{abs}(Y) = \sigma_{abs}(X + Y)$	DAT3		
$\sigma_{abs}(X)\cdot\dot{\delta} = \sigma_{abs}(X\cdot\dot{\delta})$	DAT4	$a = fts(a) + \sigma_{abs}(a)$	ADAT
$\sigma_{abs}(X)\cdot(fts(a) + Y) = \sigma_{abs}(X)\cdot Y$	DAT5		
$\sigma_{abs}(X)\cdot(fts(a)\cdot Y + Z) = \sigma_{abs}(X)\cdot Z$	DAT6		
$\sigma_{abs}(X)\cdot\sigma_{abs}(Y) = \sigma_{abs}(X\cdot\lvert Y\rvert)$	DAT7		

TABLE 10. $BPA_{dat} = BPA_\delta^\bullet$ + DAT1-7, AV1,2,3,4,8,9, ADAT.

4.2 STRUCTURED OPERATIONAL SEMANTICS.

The operational rules are more complicated in this case, as we have to keep track of which time slice we are in, we have to keep track of the global clock. $\langle x,n\rangle$ denotes x in the $(n+1)$st time slice. We have:

- if $\langle x,n\rangle \xrightarrow{\sigma} \langle x',n'\rangle$, then $x \equiv x'$, $n' = n+1$
- if $\langle x,n\rangle \xrightarrow{a} \langle x',n'\rangle$ or $\langle x,n\rangle \xrightarrow{a} \langle\surd,n'\rangle$, then $n' = n$.

The operational rules for the absolute value operator are trivial.

$$\langle \mathrm{fts}(a), 0\rangle \xrightarrow{a} \langle \surd, 0\rangle \qquad \mathrm{ID}(\langle \dot{\delta}, n\rangle) \qquad \mathrm{ID}(\langle \mathrm{fts}(a), n+1\rangle) \qquad \langle a, n\rangle \xrightarrow{a} \langle \surd, n\rangle$$

$$\langle a, n\rangle \xrightarrow{\sigma} \langle a, n+1\rangle \qquad \langle \delta, n\rangle \xrightarrow{\sigma} \langle \delta, n+1\rangle$$

$$\frac{\neg\Box\mathrm{ID}(\langle x,0\rangle)}{\langle \sigma_{\mathrm{abs}}(x),0\rangle \Box\xrightarrow{\sigma} \Box\langle \sigma_{\mathrm{abs}}(x),1\rangle} \qquad \frac{\mathrm{ID}(\langle x,n\rangle)}{\mathrm{ID}(\langle \sigma_{\mathrm{abs}}(x),n+1\rangle)} \qquad \frac{\langle x,n\rangle \Box\xrightarrow{\sigma} \Box\langle x,n+1\rangle}{\langle \sigma_{\mathrm{abs}}(x),n+1\rangle \Box\xrightarrow{\sigma} \Box\langle \sigma_{\mathrm{abs}}(x),n+2\rangle}$$

$$\frac{\langle x,n\rangle \Box\xrightarrow{a} \Box\langle x',n\rangle}{\langle \sigma_{\mathrm{abs}}(x),n+1\rangle \Box\xrightarrow{a} \Box\langle \sigma_{\mathrm{abs}}(x'),n+1\rangle} \qquad \frac{\langle x,n\rangle \Box\xrightarrow{a} \Box\langle \surd,n\rangle}{\langle \sigma_{\mathrm{abs}}(x),n+1\rangle \Box\xrightarrow{a} \Box\langle \surd,n+1\rangle}$$

$$\frac{\langle x,n\rangle \Box\xrightarrow{a} \Box\langle x',n\rangle}{\langle x+y,n\rangle \Box\xrightarrow{a} \Box\langle x',n\rangle, \Box\langle y+x,n\rangle \blacktriangleright\Box\xrightarrow{a}\langle x',n\rangle} \qquad \frac{\langle x,n\rangle \Box\xrightarrow{a} \Box\langle \surd,n\rangle}{\langle x+y,n\rangle \Box\xrightarrow{a} \Box\langle \surd,n\rangle, \Box\langle y+x,n\rangle \blacktriangleright\Box\xrightarrow{a}\langle \surd,n\rangle}$$

$$\frac{\langle x,n\rangle \Box\xrightarrow{\sigma} \Box\langle x,n+1\rangle}{\langle x+y,n\rangle \Box\xrightarrow{\sigma} \Box\langle x+y,n+1\rangle, \Box\langle y+x,n\rangle \blacktriangleright\Box\xrightarrow{\sigma}\langle y+x,n+1\rangle} \qquad \frac{\mathrm{ID}(\langle x,n\rangle), \Box\mathrm{ID}(\langle y,n\rangle)}{\mathrm{ID}(\langle x+y,n\rangle)}$$

$$\frac{\langle x,n\rangle \Box\xrightarrow{a} \Box\langle x',n\rangle}{\langle x\cdot y,n\rangle \Box\xrightarrow{a} \Box\langle x'\cdot y,n\rangle} \qquad \frac{\langle x,n\rangle \Box\xrightarrow{\sigma} \Box\langle x,n+1\rangle}{\langle x\cdot y,n\rangle \Box\xrightarrow{\sigma} \Box\langle x\cdot y,n+1\rangle} \qquad \frac{\langle x,n\rangle \Box\xrightarrow{a} \Box\langle \surd,n\rangle}{\langle x\cdot y,n\rangle \Box\xrightarrow{a} \Box\langle y,n\rangle} \qquad \frac{\mathrm{ID}(\langle x,n\rangle)}{\mathrm{ID}(\langle x\cdot y,n\rangle)}$$

TABLE 11. Operational semantics of $\mathrm{BPA}_{\mathrm{dat}}$.

4.3 BISIMULATION.

We also have to adapt the definition of bisimulation. A *bisimulation* is a symmetric binary relation R on $P \times \mathbb{N}$ such that ($a \in A$):

i. whenever $R(\langle s, n\rangle, \langle s', n'\rangle)$ then $n = n'$ and $\mathrm{ID}(\langle s, n\rangle)$ iff $\mathrm{ID}(\langle s', n'\rangle)$

ii. whenever $R(\langle s, n\rangle, \langle t, n\rangle)$ and $\langle s,n\rangle \xrightarrow{a} \langle s',n\rangle$, then there is a process expression t' such that $\langle t,n\rangle \xrightarrow{a} \langle t',n\rangle$ and $R(\langle s', n\rangle, \langle t', n\rangle)$

iii. whenever $R(\langle s, n\rangle, \langle t, n\rangle)$ and $\langle s,n\rangle \xrightarrow{a} \langle \surd,n\rangle$, then $\langle t,n\rangle \xrightarrow{a} \langle \surd,n\rangle$

iv. whenever $R(\langle s, n\rangle, \langle t, n\rangle)$ and $\langle s,n\rangle \xrightarrow{\sigma} \langle s,n+1\rangle$, then $\langle t,n\rangle \xrightarrow{\sigma} \langle t,n+1\rangle$ and $R(\langle s,n+1\rangle, \langle t,n+1\rangle)$

We say process expressions x and y are *bisimilar*, denoted $x \leftrightarrows y$, if there exists a bisimulation with $R(\langle x,0\rangle, \langle y,0\rangle)$. Bisimulation is a congruence relation on these transition systems.

4.4 BASIC TERMS.

We define the notion of a basic term as in section 3. The set of basic terms is B_{dat}, and we have auxiliary sets B_{dat}^1 (basic terms that start in the first time slice) and B_{dat}^∞ (basic terms that start in an arbitrary time slice).

1. $B_{\mathrm{dat}}^1 \subseteq B_{\mathrm{dat}}$
2. $B_{\mathrm{dat}}^\infty \subseteq B_{\mathrm{dat}}$
3. if $a \in A$, then $\mathrm{fts}(a) \in B_{\mathrm{dat}}^1$
4. $\delta \in B_{\mathrm{dat}}^\infty$
5. if $a \in A$ and $t \in B_{\mathrm{dat}}$, then $\mathrm{fts}(a)\cdot t \in B_{\mathrm{dat}}^1$
6. if $a \in A$, then $a \in B_{\mathrm{dat}}^\infty$ and $a\cdot\dot{\delta} \in B_{\mathrm{dat}}^\infty$

7. if $t,s \in B^1_{dat}$, then $t+s \in B^1_{dat}$
8. if $a \in A$ and $t \in B^\infty_{dat}$, then $a \cdot t \in B^\infty_{dat}$
9. $\dot{\delta} \in B_{dat}$
10. if $t,s \in B^\infty_{dat}$, then $t+s \in B^\infty_{dat}$
11. if $t \in B_{dat}$, then $\sigma_{abs}(t) \in B_{dat}$
12. if $t \in B^1_{dat}$ and $s \in B_{dat}$, then $t + \sigma_{abs}(s) \in B_{dat}$.

Basic terms over BPA_{dat} are defined by clauses 1-12; we obtain basic terms over BPA^-_{dat} by omitting clauses 2,4,6,8,10. Notice the different formulation in clauses 6 and 8. As an example, consider the following calculation: $a \cdot \sigma_{abs}(fts(b)) = fts(a) \cdot \sigma_{abs}(fts(b)) + \sigma_{abs}(fts(a) \cdot fts(b) + \sigma_{abs}(a\dot{\delta}))$

Again, the basic terms over BPA^-_{dat} are exactly the terms of one of the following three forms:

a. $\dot{\delta}$

b. $\sum_{i<n} \Box fts(a_i) \cdot t_i + \sum_{j<m} \Box fts(b_j)$, with $n+m>0$, $a_i, b_j \in A$ and t_i basic;

c. $\sum_{i<n} \Box fts(a_i) \cdot t_i + \sum_{j<m} \Box fts(b_j) + \sigma_{abs}(t)$, with $n,m \geq 0$, $a_i, b_j \in A$ and t, t_i basic.

In the case of BPA_{dat}, we can have extra summands of the form a ($a \in A_\delta$) or $a \cdot t$ in form b. As in 3.4-6, we obtain elimination, soundness and completeness. As in 3.7, we can define a graph model.

4.5 PROJECTION.

We can define projections as in 3.8. We give axioms in table 12 below. As before, we can define a projective limit model. Again, we have elimination, soundness and completeness.

$\pi^{abs}_0(X) = \dot{\delta}$	PRA1	$\pi^{abs}_{n+1}(fts(a)) = fts(a)$	PRA4
$\pi^{abs}_n(\dot{\delta}) = \dot{\delta}$	PRA2	$\pi^{abs}_{n+1}(fts(a) \cdot X) = fts(a) \cdot \pi^{abs}_n(X)$	PRA5
$\pi^{abs}_{n+1}(\sigma_{abs}(X)) = \sigma_{abs}(\pi^{abs}_n(X))$	PRA3	$\pi^{abs}_n(X + Y) = \pi^{abs}_n(X) + \pi^{abs}_n(Y)$	PRA6

TABLE 12. Projections.

4.6 FREE MERGE.

The addition of parallel composition is analogous to the relative time case. We show the axioms in table 13, the operational rules in table 14. We have elimination, soundness, completeness as before. A sample calculation: $fts(a) \parallel fts(b) \cdot \dot{\delta} = fts(a) \cdot fts(b) \cdot \dot{\delta} + fts(b) \cdot \dot{\delta}$.

$fts(a) \between (X + fts(\delta)) = fts(a) \cdot (X + fts(\delta))$	DATCM2
$fts(a) \cdot X \between (Y + fts(\delta)) = fts(a) \cdot (X \parallel (Y + fts(\delta)))$	DATCM3
$\sigma_{abs}(X) \between (fts(a) + Y) = \sigma_{abs}(X) \between (fts(\delta) + Y)$	DAT8
$\sigma_{abs}(X) \between (fts(a) \cdot Y + Z) = \sigma_{abs}(X) \between (fts(\delta) + Z)$	DAT9
$\sigma_{abs}(X) \between \sigma_{abs}(Y) = \sigma_{abs}(X \between Y)$	DAT10

TABLE 13. $PA_{dat} = BPA_{dat}$ + M1, CM4, LMID1,2, DATCM2,3, CM2AV, CM3AV, AV5, DAT8-10.

$$\frac{\langle x,n\rangle \Box\!\!\xrightarrow{a} \Box\langle x',n\rangle, \Box\neg\Box \mathrm{ID}(\langle y,n\rangle)}{\langle x \between y,n\rangle \Box\!\!\xrightarrow{a} \Box\langle x' \between y,n\rangle, \Box\langle y \between x,n\rangle \boxdot\!\!\Box\!\!\xrightarrow{a} \Box\langle y \between x',n\rangle, \Box\langle x \between y,n\rangle \boxdot\!\!\Box\!\!\xrightarrow{a} \Box\langle x' \between y,n\rangle}$$

$$\frac{\langle x,n\rangle \Box\!\!\xrightarrow{a} \Box\langle \surd,n\rangle, \Box\neg\Box \mathrm{ID}(\langle y,n\rangle)}{\langle x \between y,n\rangle \Box\!\!\xrightarrow{a} \Box\langle y,n\rangle, \Box\langle y \between x,n\rangle \boxdot\!\!\Box\!\!\xrightarrow{a} \langle y,n\rangle, \Box\langle x \between y,n\rangle \boxdot\!\!\Box\!\!\xrightarrow{a} \langle y,n\rangle}$$

$$\frac{\langle x,n\rangle \Box\!\!\xrightarrow{\sigma} \Box\langle x,n+1\rangle, \Box\langle y,n\rangle \boxdot\!\!\Box\!\!\xrightarrow{\sigma} \langle y,n+1\rangle}{\langle x \between y,n\rangle \Box\!\!\xrightarrow{\sigma} \Box\langle x \between y,n+1\rangle, \Box\langle x \between y,n\rangle \boxdot\!\!\Box\!\!\xrightarrow{\sigma} \langle x \between y,n+1\rangle}$$

$$\frac{\mathrm{ID}(\langle x,n\rangle)}{\mathrm{ID}(\langle x \between y,n\rangle), \Box\mathrm{ID}(\langle y \between x,n\rangle), \Box\mathrm{ID}(\langle x \between y,n\rangle), \Box\mathrm{ID}(\langle y \between x,n\rangle)}$$

TABLE 14. Operational semantics of PA_{dat}.

4.7 ALGEBRA OF COMMUNICATING PROCESSES.

The addition of communication is analogous to the relative time case. We show the axioms in table 15, the operational rules in table 16. We have elimination, soundness, completeness as in the case of relative time.

$\mathrm{fts}(a) \mid \mathrm{fts}(b) = \mathrm{fts}(a \mid b)$	DATC	$\mathrm{fts}(a) \between \sigma_{abs}(X) = \mathrm{fts}(\delta)$	DAT11
$\mathrm{fts}(a)\cdot X \between \mathrm{fts}(b) = \mathrm{fts}(a \between b)\cdot X$	DATCM5	$\sigma_{abs}(X) \between \mathrm{fts}(a) = \mathrm{fts}(\delta)$	DAT12
$\mathrm{fts}(a) \between \mathrm{fts}(b)\cdot X = \mathrm{fts}(a \mid b)\cdot X$	DATCM6	$\mathrm{fts}(a)\cdot X \between \sigma_{abs}(Y) = \mathrm{fts}(\delta)$	DAT13
$\mathrm{fts}(a)\cdot X \between \mathrm{fts}(b)\cdot Y = \mathrm{fts}(a \between b)\cdot(X \between Y)$	DATCM7	$\sigma_{abs}(X) \between \mathrm{fts}(a)\cdot Y = \mathrm{fts}(\delta)$	DAT14
$\partial_H(\mathrm{fts}(a)) = \mathrm{fts}(a)$ if $a \notin H$	DATD1	$\sigma_{abs}(X) \between \sigma_{abs}(Y) = \sigma_{abs}(X \between Y)$	DAT15
$\partial_H(\mathrm{fts}(a)) = \mathrm{fts}(\delta)$ if $a \in H$	DATD2	$\partial_H(\sigma_{abs}(X)) = \sigma_{abs}(\partial_H(X))$	DAT16

TABLE 15. $ACP_{dat} = PA_{dat}$ - M1 + $ACP^{\bullet}_{\delta}$ + DATC, DATCM5-7, DRTD1,2, DAT11-16.

$$\frac{\langle x,n\rangle \xrightarrow{a} \surd\langle x',n\rangle, \surd\langle y,n\rangle \xrightarrow{b} \surd\langle y',n\rangle, \surd a \mid b=c}{\langle x \parallel y,n\rangle \xrightarrow{c} \surd\langle x' \parallel y',n\rangle, \surd\langle x \mid y,n\rangle \xrightarrow{c} \surd\langle x' \parallel y',n\rangle} \qquad \frac{ID(\langle x,n\rangle)}{ID(\langle x \mid y,n\rangle), \surd ID(\langle y \mid x,n\rangle)}$$

$$\frac{\langle x,n\rangle \xrightarrow{a} \surd\langle x',n\rangle, \surd\langle y,n\rangle \xrightarrow{b} \surd\langle \surd,n\rangle, \surd a \mid b=c}{\langle x \parallel y,n\rangle \xrightarrow{c} \surd\langle x',n\rangle, \surd\langle y \parallel x,n\rangle \xrightarrow{c} \surd\langle x',n\rangle, \surd\langle x \mid y,n\rangle \xrightarrow{c} \surd\langle x',n\rangle, \surd\langle y \mid x,n\rangle \xrightarrow{c} \surd\langle x',n\rangle}$$

$$\frac{\langle x,n\rangle \xrightarrow{a} \surd\langle \surd,n\rangle, \surd\langle y,n\rangle \xrightarrow{b} \surd\langle \surd,n\rangle, \surd a \mid b=c}{\langle x \parallel y,n\rangle \xrightarrow{c} \surd\langle \surd,n\rangle, \surd\langle x \mid y,n\rangle \xrightarrow{c} \surd\langle \surd,n\rangle} \qquad \frac{\langle x,n\rangle \xrightarrow{\sigma} \surd\langle x,n+1\rangle, \surd\langle y,n\rangle \xrightarrow{\sigma} \surd\langle y,n+1\rangle}{\langle x \mid y,n\rangle \xrightarrow{\sigma} \surd\langle x \mid y,n+1\rangle}$$

$$\frac{\langle x,n\rangle \xrightarrow{a} \surd\langle x',n\rangle, \surd a \notin H}{\langle \partial_H(x),n\rangle \xrightarrow{a} \surd\langle \partial_H(x'),n\rangle} \qquad \frac{\langle x,n\rangle \xrightarrow{\sigma} \surd\langle x,n+1\rangle}{\langle \partial_H(x),n\rangle \xrightarrow{\sigma} \surd\langle \partial_H(x),n+1\rangle} \qquad \frac{\langle x,n\rangle \xrightarrow{a} \surd\langle \surd,n\rangle, \surd a \notin H}{\langle \partial_H(x),n\rangle \xrightarrow{a} \surd\langle \surd,n\rangle}$$

$$\frac{ID(\langle x,n\rangle)}{ID(\langle \partial_H(x),n\rangle)}$$

TABLE 16. Additional operational rules of ACP_{dat}.

5. PARAMETRIC TIME.

In this section we will integrate the absolute time and the relative time approach. Our aim is to present a finite axiomatization, that allows an elimination theorem. As a consequence, we can expand expressions like $\mathsf{cts}(a) \parallel \mathsf{fts}(b)$, $\mathsf{cts}(a) \parallel (\mathsf{fts}(b) + \delta)$. In [BAB92], we used a different approach, using variable binding, that did not allow a finite axiomatization.

We introduce two new operators: Θ, the (relative) time spectrum combinator (comparable to the push operator on stacks or the cons operator on lists), and μ, the spectrum tail operator (comparable to a pop, tail or rest operator). The absolute value operator introduced in 2.1 can also be called the spectrum head operator. $P \Theta Q$ is a process that when initialised in the first time slice behaves as $|P|$; when initialised in slice $n+1$ its behaviour is determined by Q as follows: initialise in slice n thereafter apply σ_{abs}. $\mu(X)$ computes a process such that $X = |X| \Theta \mu(X)$. For a parametric discrete time process we have the time spectrum sequence $|X|$, $|\mu(X)|$, $|\mu^2(X)|$, ... For each infinite sequence $(P_n)_{n \in \mathbb{N}}$ one may imagine a process P with $|\mu^n(P)| = P_n$ though not all such P can be finitely expressed.

5.1 BASIC PROCESS ALGEBRA.

We put together the theories BPA_{drt}^{-} and BPA_{dat}^{-}. We have presented all axioms in such a way, that they remain valid on the extended syntax. In axioms CM2AV, CM3AV, DAT7, we needed the absolute value operator, that characterizes absolute timing processes. On the other hand, the spectrum tail operator characterizes relative timing processes. The theory BPA_{dpt}^{-} unites BPA_{drt}^{-} and BPA_{dat}^{-} and the

additional axioms AV10-12, ST1-7, SC1-3. BPA_{dpt} adds constants $A \cup \{\delta\}$ and axioms A6A, A7, AV2, ADRT, ADAT. Notice that ST6,7 imply $\mu(a) = a$.

$\sigma_{abs}(X) = \sigma_{abs}(\lvert X \rvert)$	AV10	$\mu(\dot{\delta}) = \dot{\delta}$	ST1
$\lvert cts(a) \rvert = fts(a)$	AV11	$\mu(cts(a)) = cts(a)$	ST2
$\lvert \sigma_{rel}(X) \rvert = \sigma_{abs}(\lvert \mu(X) \rvert)$	AV12	$\mu(X + Y) = \mu(X) + \mu(Y)$	ST3
		$\mu(X \cdot Y) = \mu(X) \cdot \mu(Y)$	ST4
$\lvert X \trianglerighteq Y \rvert = \lvert X \rvert$	SC1	$\mu(\sigma_{rel}(X) = \sigma_{rel}(\mu(X))$	ST5
$\mu(X \trianglerighteq Y) = Y$	SC2	$\mu(fts(a)) = \dot{\delta}$	ST6
$X = \lvert X \rvert \trianglerighteq \mu(X)$	SC3	$\mu(\sigma_{abs}(X)) = \lvert X \rvert$	ST7

TABLE 17. $BPA_{dpt} = BPA_{drt} + BPA_{dat}$ + AV10-12, ST1-7, SC1-3.

Note that the three axioms DAT5-7 can be replaced by $\sigma_{abs}(X) \cdot Y = \sigma_{abs}(X \cdot \mu(Y))$. We can define:

- P is an absolute time process iff $BPA_{dpt} \vdash P = \lvert P \rvert$
- P is an relative time process iff $BPA_{dpt} \vdash P = \mu(P)$.

We see that $BPA^{\bullet}_{\delta}$ constitutes the intersection of the absolute time and relative time theories.

5.2 BASIC FORM.

We claim that each $BPA_{\overline{dpt}}$ (resp. BPA_{dpt}) process expression can be written in the form

$$P_1 \trianglerighteq P_2 \trianglerighteq \ldots P_n \trianglerighteq Q$$

(we omit brackets, using the convention that $\trianglerighteq$ associates to the right), such that each P_i is a $BPA_{\overline{dat}}$-basic term (resp. BPA_{dat}-basic term) and Q is a $BPA_{\overline{drt}}$-basic term (resp. BPA_{drt}-basic term).

The way we achieve this is by writing

$$X = \lvert X \rvert \trianglerighteq \lvert \mu(X) \rvert \trianglerighteq \ldots \lvert \mu^n(X) \rvert \trianglerighteq \mu^{n+1}(X).$$

Now one can reduce each $\lvert \mu^n(X) \rvert$ to a $BPA_{\overline{dat}}$-basic term and if n is sufficiently large, we can write $\mu^{n+1}(X))$ without any σ_{abs} or $fts(a)$ using ST1-7, so will be in the relative time signature.

We call $(\lvert X \rvert, \lvert \mu(X) \rvert, \lvert \mu^2(X) \rvert, \ldots)$ the time spectrum expansion sequence (TSS) of X. The nth component is called $tss_n(X)$.

The Extensionality principle for Parametric Discrete Time EPDT is the principle that processes are equal if they have the same TSS. In this way, each model for absolute time processes induces a model for parametric time processes. We can prove that our axiomatisation is complete for this model.

Some examples of time spectrum sequences:

The time spectrum sequence of $cts(a)$ is $(fts(a), fts(a), fts(a), \ldots)$

The time spectrum sequence of $fts(a)$ is $(fts(a), \dot{\delta}, \dot{\delta}, \ldots)$

The time spectrum sequence of $\sigma_{abs}(fts(a))$ is $(\sigma_{abs}(fts(a)), fts(a), \dot{\delta}, \dot{\delta}, \ldots)$.

The nth component tss_n distributes over $+, \between, \lfloor\!\lfloor, \lfloor, \partial_H$. Furthermore $tss_n(X \cdot Y) = tss_n(X) \cdot \mu^n(Y)$.

5.3 EXAMPLE.

$\sigma_{rel}(fts(a)) = \lvert \sigma_{rel}(fts(a)) \rvert \trianglerighteq \mu(\sigma_{rel}(fts(a))) = \sigma_{abs}(\lvert \mu(fts(a)) \rvert) \trianglerighteq \sigma_{rel}(\mu(fts(a))) = \sigma_{abs}(\dot{\delta}) \trianglerighteq \sigma_{rel}(\dot{\delta})$
$= fts(\delta) \trianglerighteq cts(\delta)$ (this is the basic form) $= \lvert cts(\delta) \rvert \trianglerighteq \mu(cts(\delta)) = cts(\delta)$.

5.4 PARALLEL COMPOSITION.

The extension to PA or ACP is straightforward. The theory $PA_{\overline{dpt}}$ unites $BPA_{\overline{dpt}}$, $PA_{\overline{drt}}$ and $PA_{\overline{dat}}$ and the additional axioms ST8,9 of table 18. PA_{dpt} unites BPA_{dpt}, PA_{drt} and PA_{dat} and ST8,9. $ACP_{\overline{dpt}}$ unites $BPA_{\overline{dpt}}$, $ACP_{\overline{drt}}$ and $ACP_{\overline{dat}}$ and ST8-11. ACP_{dpt} unites BPA_{dpt}, ACP_{drt} and ACP_{dat} and ST8-11.

$\mu(X \between Y) = \mu(X) \between \mu(Y)$	ST8	$\mu(X \mid Y) = \mu(X) \mid \mu(Y)$	ST10
$\mu(X \between Y) = \mu(X) \between \mu(Y)$	ST9	$\mu(\partial_H(X)) = \partial_H(\mu(X))$	ST11

TABLE 18. Extension to $PA_{\overline{dpt}}$ and $ACP_{\overline{dpt}}$.

We can again achieve the basic form of 6.2. As a result, we achieve elimination, soundness and completeness. An example of the elimination of the merge (no communication):

$(\sigma_{abs}(fts(a)) + \sigma_{rel}^2(cts(b))) \between cts(c) = fts(c)\cdot\sigma_{abs}(fts(a)\cdot\sigma_{abs}(fts(b)))\Theta$
$(\sigma_{abs}((fts(a)\cdot fts(c) + fts(c)\cdot fts(a))\cdot\sigma_{abs}(fts(b)))\Theta cts(c)\cdot\sigma_{rel}(cts(b)))$.

5.5 EXTENSIONALITY.

We formulate initialisation in an arbitrary time slice following [BAB92] by introducing natural numbers with 0 and successor (succ). $n \gg X$ denotes the behavior of X upon initialisation at absolute time n. We have the following defining equations. As is usual, we write $n+1$ for succ(n), $n+2$ for succ(succ(n)) and so on.

$0 \gg X = |X|$ $\qquad$ $n+1 \gg X = \sigma_{abs}(n \gg \mu(X))$.

Now we have the following alternative expression for extensionality for parametric discrete time:

$$\frac{\text{for all } n, n \gg P = n \gg Q}{P = Q} \qquad \text{EPDT.}$$

It is the intended meaning of (parametric) discrete time process algebra that EPDT is satisfied. We note that with the use of EPDT one can prove elimination and completeness in the case of ACP_{dpt} without the use of μ. Following [BAB93] one may use the following notation: if F is a process expression depending on x, then $\surd_d x.\ F$ is the parametric time process defined by $n \gg (\surd_d x.\ F) = n \gg F[n/x]$. We call $\surd_d x.\ F$ the initial abstraction operator for discrete time.

5.6 PARAMETRIC TIME CONDITIONS.

Another way to denote extensionality is obtained if we introduce parametric time conditions. We consider the following syntax.

- sort B_{par} — sort of parametric time Booleans
- constants true, false — standard Booleans
- functions $\neg : B_{par} \to B_{par}$ — negation
  $\wedge, \vee : B_{par} \times B_{par} \to B_{par}$ — conjunction, disjunction
  $|_| : B_{par} \to B_{par}$ — spectrum head
  $\mu : B_{par} \to B_{par}$ — spectrum tail
  $\Theta : B_{par} \times B_{par} \to B_{par}$ — spectrum combinator.

We have the axioms shown in table 18. They constitute the theory $BOOL_{dpt}$. We link parametric time booleans to parametric time processes by means of the conditional operator $.\lhd.\rhd.$ $X \lhd b \rhd Y$ represents if b then X else Y (notation taken from [HHJ+87]).

$\neg$ true = false	BOOL1	$b \vee c = c \vee b$	BOOL4
$\neg$ false = true	BOOL2	true $\vee$ b = true	BOOL5
$\neg \neg b = b$	BOOL3	false $\vee$ b = b	BOOL6
$X \lhd$ true $\rhd Y = X$	BOOL7	$b \wedge c = \neg (\neg b \vee \neg c)$	BOOL9
$X \lhd$ false $\rhd Y = Y$	BOOL8		
$\|true\| = true$	BAV1	μ(true) = true	BST1
$\|false\| = false$	BAV2	μ(false) = false	BST2
$\|\neg b\| = \neg \|b\|$	BAV3	$\mu(\neg b) = \neg \mu(b)$	BST3
$\|b \vee c\| = \|b\| \vee \|c\|$	BAV4	$\mu(b \vee c) = \mu(b) \vee \mu(c)$	BST4
$\|X \lhd b \rhd Y\| = \|X\| \lhd \|b\| \rhd \|Y\|$	BAV5	$\mu(X \lhd b \rhd Y) = \mu(X) \lhd \mu(b) \rhd \mu(Y)$	BST5
$\|b \Theta c\| = \|b\|$	BSC1	$\mu(b \Theta c) = c$	BSC2
$b = \|b\| \Theta \mu(b)$	BSC3		
$\neg (b \Theta c) = \neg b \Theta \neg c$	BSC4	$(b \Theta c) \vee (b' \Theta c') = (b \vee b') \Theta (c \vee c')$	BSC5

TABLE 18. $BOOL_{dpt}$ = BOOL1-9, BAV1-5, BST1-5, BSC1-5.

Using this syntax, we can define the parametric time Booleans $sl(n)$ (true iff the current time slice is the nth slice), $sl_>(n)$ (true iff the current slice is larger than n) and the initialisation operator as follows:

$sl(0) = false$ $\quad sl(1) = true \Theta false$ $\quad sl(n+2) = false \Theta sl(n+1)$

$sl_>(0) = true$ $\quad sl_>(n+1) = false \Theta sl_>(n)$

$0 \gg b = |b|$ $\quad n+1 \gg b = n \gg \mu(b)$.

It follows that $n \gg (X \lhd b \rhd Y) = n \gg X \lhd n \gg b \rhd n \gg Y$. A useful abbreviation is $b ::\!\!\rightarrow X$ for $X \lhd b \rhd \dot{\delta}$. Now extensionality can be written as the time spectrum expansion TSE as follows:

$$X = \sum_{n=0}^{\infty} \Box sl(n+1) \Box ::\!\!\rightarrow \Box (n \Box \gg \Box X) \qquad \text{TSE},$$

and approximated with a finite sum, TSE_k:

$$X = \sum_{n=0}^{k-1} \Box sl(n+1) \Box ::\!\!\rightarrow \Box (n \Box \gg \Box X) + sl_>(k) ::\!\!\rightarrow X.$$

We call the last summand $sl_>(k) ::\!\!\rightarrow X$ the residue of X after slice k. We notice that for X a closed expression in the syntax of ACP_{dpt}, from some k onward the residue is (extensionally) equal to a process which is the residue of a relative time process. This allows yet another way to obtain basic forms and elimination results for parametric time processes.

An interesting additional operator is the operator ψ defined by equations

$\psi(true \Theta X) = false \Theta true \Theta \psi(X)$ $\quad \psi(false \Theta X) = false \Theta \psi(X)$.

Using this operator, all eventually periodic time conditions can be expressed. This is useful in the representation of regular parametric time processes.

5.7 PROJECTION.
We can extend the definition of the absolute time projection operators to parametric time processes by adding to the axioms in 4.5 (table 11) the axiom $\pi^{abs}_{n+1}(X \odot Y) = \pi^{abs}_{n+1}(|X|) \odot \pi^{abs}_{n}(Y)$. For a relative time process P, the relationship between the relative time and absolute time projection operators is given by the axiom $n \gg \pi^{rel}_{k}(P) = \pi^{abs}_{n+k}(n \gg P)$.

5.8 RECURSION.
If P is a closed process expression, then we may view $X = P \odot X$ as a guarded equation in X. In many models, all guarded equations will have unique solutions. As an example, $cts(a)$ is the unique solution of $X = fts(a) \odot X$. For each such equation, the solution X is a relative time process (because $\mu(X) = X$). Further, we have $|X| = |P|$. We write $|P|_{rel}$ for the solution, so we have $|P|_{rel} = |P|_{abs} \odot |P|_{rel}$. In this paper, we will not use the operator $|_|_{rel}$ and consequently, we will not consider axioms for it.

5.9 REGULAR PROCESSES.
In [BEK84b], finite state processes modulo (strong) bisimulation were studied in the context of ACP. The definitions given there can be easily extended to discrete time.

First of all, a relative time process is regular if its operational semantics as given in 3.2,7,8 yields a transition system which is bisimilar to a finite one. Regular relative time processes can be defined using linear systems of equations over $\dot{\delta}$, $cts(a)$, $+$, σ_{rel} and $cts(a)\cdot_$, the prefix restriction of sequential composition. As an example, the linear equation $X = cts(a) + \sigma_{rel}(X)$ defines the process a.

Next, if X is a regular relative time process, then $|X|$ is regular in absolute time. This is used as a definition, i.e. an absolute time process X is regular iff for some regular relative time Y we have $X = |Y|$. To define regular absolute time processes using systems of linear equations we need $|_|$ and the detour via relative time.

Finally, a parametric time process is regular if its time spectrum sequence $(|\mu^n(X)|)_{n \in \mathbb{N}}$ is an eventually periodic sequence of regular processes.

5.10 REMARK.
Similar to the binary Kleene star operator with defining equation $X^*Y = X\cdot(X^*Y) + Y$ (see [BEBP94]) we can introduce iterated delay: $\sigma^*_{rel}(X) = \sigma_{rel}(\sigma^*_{rel}(X)) + X$. To give an example, we have $\sigma^*_{rel}(cts(a)) = a$. A very interesting area of work is to investigate axiomatisations for fragments of process algebra with iteration operators. Along the lines of [FOZ94], [FOK94] several results in the case of discrete time can be obtained.

6. RELATION TO EARLIER WORK.

The relation between the two-phase versions of discrete time presented here and the timestamped versions of [BAB92] is as follows:

$\underline{a}(n+1) = \sigma_{abs}^{n}(fts(a))$ $\quad\quad$ $\underline{a}[n+1] = \sigma_{rel}^{n}(cts(a))$.

In turn, we can relate these to timestamped real time of [BAB91] as follows:

$\underline{a}(n+1) = \int_{t\in(n,n+1)} \Box\Box\Box\Box a(t)$ $\quad\quad$ $\underline{a}[n+1] = \surd t.\int_{r\in(\Box\lfloor t\rfloor,\lfloor t\rfloor+1)} \Box\Box\Box\Box a(r)$

Many extensions of ACP have been proposed, we mention: state operator, priority operator, process creation operators, unreliable communication primitives, iteration, early input prefix, process prefix. In all cases we find that it is completely straightforward to determine timed versions in absolute, relative and parametric time.

7. COMPARISON WITH RELATED WORK.

Other discrete time process algebras that we encountered all use relative timing and are two-phase. None of them is related to a time free theory. Therefore, we limit ourselves to comparing other theories to ACP_{drt}.

Of course, any process algebra involving multi-way synchronisation can be used as a discrete time process algebra, by just designating a special atomic action as a clock tick. This is the approach followed in CIRCAL [MIL85], and indeed we can do this in ACP as well. In the following, we consider algebras where passage of time has properties different from atomic actions.

In BPA_{drt}, we combine a unary operator for (unit) delay like in TCCS [MOT89], time factorisation as in ATP [NIS94], TCCS [MOT89], TPCCS [HAN91] and TPL [HER90], the interpretation of + as the weak choice of TCCS, but we notice a significant difference with TPL [HER90] because visible actions cannot idle (here, we follow TCCS, ACP_t and ATP).

Our definition of free merge in PA_{drt} is similar to the definitions in TCCS, ATP, ACP_t and TPL (the correspondence with the TPL definition holds in the absence of communication; in the presence of communication TPL's merge will give priority to internal communications). So it follows that without communication the relation between merge and discrete time is not controversial. We depart from ATP by allowing time stops. Indeed, in our setting $\sigma_{rel}(cts(a)) \between cts(\delta)$ is unable to perform a σ-step.

The merge of ACP_{drt} works just as the merge of TCCS [MOT89] (taking weak choice of TCCS for +). It is also equivalent to the merge of ATP. It differs from merge in [GRO90a], set up in ACP_t. In fact, ACP_t contains axioms $\sigma \between x = \delta$ and $\sigma x \between y = \delta$. Our main objection against these axioms (which occur in ATP as well) is that they render it impossible to injectively embed ACP_t into $ACP\rho$ of [BAB91] or its extension $ACP\rho\surd\Box\Box$ of [BAB93].

It follows that the only disagreement in the literature is about the proper semantics of left merge and communication merge. Therefore, our choice for the merge operator itself seems reasonably well motivated.

8. CONCLUDING REMARKS.

We have proposed a theory on discrete time process algebra that extends ACP with the features of ATP in a setting consistent with that of ACPρ, ACPρ√, ACPρ√I. Many options for further work remain. We mention some:

i. All extensions of ACP that have been developed can be integrated in the discrete time setting, such as state operators, process creation, signals, priorities, asynchronous communication (see [BAW90] for these ACP extensions).

ii. The analysis of $ACP_{dp\rho}$ as a term rewriting system may be worth attention.

iii. Like in the case of ACP, many more models than the bisimulation model are possible. We sketched a projective limit model. Investigation of other models such as failures, ready, trace models can be worth while.

REFERENCES.

[BAB91] J.C.M. BAETEN & J.A. BERGSTRA, *Real time process algebra,* Formal Aspects of Computing 3 (2), 1991, pp. 142-188.

[BAB92] J.C.M. BAETEN & J.A. BERGSTRA, *Discrete time process algebra.* Extended abstract in: Proc. CONCUR'92, Stony Brook (W.R. Cleaveland, ed.), LNCS 630, Springer 1992, pp. 401-420. Full version: report P9208b, Programming Research Group, University of Amsterdam 1992, pp. 81. Available at ftp://ftp.win.tue.nl/pub/techreports/josb/discretetime.ps.Z.

[BAB93] J.C.M. BAETEN & J.A. BERGSTRA, *Real space process algebra,* Formal Aspects of Computing 5 (6), 1993, pp. 481-529.

[BAB94] J.C.M. BAETEN & J.A. BERGSTRA, *On sequential composition, action prefixes and process prefix,* Formal Aspects of Computing 6 (3), 1994, pp. 250-268.

[BAW90] J.C.M. BAETEN & W.P. WEIJLAND, *Process algebra,* Cambridge Tracts in Theor. Comp. Sci. 18, Cambridge University Press 1990.

[BEBP94] J.A. BERGSTRA, I. BETHKE & A. PONSE, *Process algebra with iteration and nesting,* The Computer Journal 37 (4), 1994, pp. 243-258.

[BEK84a] J.A. BERGSTRA & J.W. KLOP, *Process algebra for synchronous communication,* Inf. & Control 60, 1984, pp. 109-137.

[BEK84b] J.A. BERGSTRA & J.W. KLOP, *The algebra of recursively defined processes and the algebra of regular processes.* Extended abstract in: Proceedings 11th ICALP, Antwerpen 1984 (J. Paredaens, ed.), Springer LNCS 172, 1984, pp. 82-95. Full version in: Proc. ACP94 (A. Ponse, C. Verhoef & S.F.M. van Vlijmen, eds.), Utrecht 1994, Workshops in Computing, Springer Verlag 1995, pp. 1-25.

[BEK94] J.A. BERGSTRA & P. KLINT, *The toolbus - a component interconnection architecture -,* report P9408, Programming Research Group, University of Amsterdam 1994.

[BEK95] J.A. BERGSTRA & P. KLINT, *The discrete time toolbus*, report P95xx, Programming Research Group, University of Amsterdam 1995.

[BEM95] J.A. BERGSTRA & C.A. MIDDELBURG, *A process algebra semantics for ϕSDL,* report LGPS xx, Dept. of Philosophy, Utrecht University 1995.
[BRHR84] S.D. BROOKES, C.A.R. HOARE & A.W. ROSCOE, *A theory of communicating sequential processes,* Journal of the ACM 31 (3), 1984, pp. 560-599.
[FOK94] W.J. FOKKINK, *A complete equational axiomatisation for prefix iteration,* IPL 52 (6), 1994, pp. 333-337.
[FOZ94] W.J. FOKKINK & H. ZANTEMA, *Basic process algebra with iteration: completeness of its equational axioms,* The Computer Journal 37 (4), 1994, pp. 259-267.
[GRO90a] J.F. GROOTE, *Specification and verification of real time systems in ACP,* in: Proc. 10th Symp. on Protocol Specification, Testing and Verification, Ottawa (L. Logrippo, R.L. Probert & H. Ural, eds.), North-Holland, Amsterdam 1990, pp. 261-274.
[GM91] D.P. GRUSKA & A. MAGGIOLO-SCHETTINI, *A timed process description language based on CCS,* report TR-9/91, Universitá di Pisa 1991.
[HAN91] H. HANSSON, *Time and probabilities in formal design of distributed systems,* Ph.D. thesis, report DoCS 91/27, Uppsala University 1991.
[HER90] M. HENNESSY & T. REGAN, *A temporal process algebra,* report 2/90, University of Sussex 1990.
[HOA85] C.A.R. HOARE, *Communicating sequential processes,* Prentice Hall 1985.
[HHJ+87] C.A.R. HOARE, I.J. HAYES, HE JIFENG, C.C. MORGAN, A.W. ROSCOE, J.W. SANDERS, I.H. SORENSEN, J.M. SPIVEY & B.A. SUFRIN, *Laws of programming,* CACM 30, 1987, pp. 672-686.
[KLU93] A.S. KLUSENER, *Models and axioms for a fragment of real time process algebra,* Ph.D. Thesis, Eindhoven University of Technology 1993.
[MIL85] G.J. MILNE, *Circal and the representation of communication, concurrency and time,* ACM TOPLAS 7, 1985.
[MIL89] R. MILNER, *Communication and concurrency,* Prentice-Hall 1989.
[MOT89] F. MOLLER & C. TOFTS, *A temporal calculus of communicating systems,* report LFCS-89-104, University of Edinburgh 1989.
[MOT90] F. MOLLER & C. TOFTS, *A temporal calculus of communicating systems,* in: Proc. CONCUR'90, Amsterdam (J.C.M. Baeten & J.W. Klop, eds.), LNCS 458, Springer 1990, pp. 401-415.
[NIS94] X. NICOLLIN & J. SIFAKIS, *The algebra of timed processes ATP: theory and application,* Information & Computation 114, 1994, pp. 131-178.
[NSVR90] X. NICOLLIN, J.-L. RICHIER, J. SIFAKIS & J. VOIRON, *ATP: an algebra for timed processes,* in Proc. IFIP TC2 Conf. on Progr. Concepts & Methods, Sea of Gallilee, Israel 1990.
[RER88] G.M. REED & A.W. ROSCOE, *A timed model for communicating sequential processes,* TCS 58, 1988, pp. 249-261.
[VER94] C. VERHOEF, *A congruence theorem for structured operational semantics with predicates and negative premises,* in: Proc. CONCUR'94, Uppsala (B. Jonsson & J. Parrow, eds.), Springer LNCS 836, 1994, pp. 433-448.