Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

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

Conditional axioms and α/β calculus in process algebra

Department of Computer Science   Report CS-R8502   February

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

CONDITIONAL AXIOMS AND α/β CALCULUS IN PROCESS ALGEBRA

J.C.M. BAETEN, J.A. BERGSTRA, J.W. KLOP

Centre for Mathematics and Computer Science, Amsterdam

We define the alphabet of finite and infinite terms in ACP_τ, the algebra of communicating processes with silent steps, and also give approximations of it (the α/β calculus).
Using the alphabet, we formulate some conditional axioms. The usefulness of these axioms is demonstrated in several examples.

69F11, 69F12, 69F32, 69F43

1980 MATHEMATICS SUBJECT CLASSIFICATION: 68B10, 68C01, 68D25, 68F20.
1982 CR CATEGORIES: F.1.1, F.1.2, F.3.2, F.4.3.
KEY WORDS & PHRASES: concurrency, process algebra, alphabet, conditional axiom.
NOTE: This report will be submitted for publication elsewhere.
This work was sponsored in part by ESPRIT contract METEOR.

Report CS-R8502
Centre for Mathematics and Computer Science
P.O.Box 4079, 1009 AB Amsterdam, The Netherlands

Introduction

Process algebra, as described in Bergstra & Klop [6], is a mathematical theory of specification and verification of concurrent systems. Process algebra is a modification of Milner's CCS [11], optimized towards mathematical manageability.

The essence of the approach is an axiomatic framework for the description of processes; verifications are to be done on the basis of the axiomatic laws. In particular, this approach will hopefully allow, that in actual system verification, one can avoid the introduction and inspection of models of concurrency. In our view, the difficulty with these models is the fact that both for a computer and for the human mind it is nontrivial to represent and analyse them. We think that the fundamental importance of models of concurrency is to establish the consistency of axioms. For these models, see De Bakker & Zucker [2], Brookes, Hoare & Roscoe [9] and Hennessy [10].

To us, the most interesting part of research is to find new axioms and to illustrate their use by means of illuminating examples. In this paper we introduce the conditional axioms CA1-7 (see § 4). Some of these axioms have a condition involving the alphabet $\alpha(X)$ of the processes involved. A typical example is:

$$\text{CA4} \quad \frac{\alpha(X)\cap I=\varnothing}{\tau_I(X)=X}$$

in words: if no action from I occurs in X, then hiding (abstracting from) actions of I in X has no effect. A more interesting axiom is:

$$\text{CA2} \quad \frac{\alpha(X)\,|\,(\alpha(Y)\cap I)=\varnothing}{\tau_I(X\|Y)=\tau_I(X\|\tau_I(Y))}.$$

This axiom allows one to commute abstraction and parallel composition (in appropriate circumstances). CA2 is absolutely vital for the verification of systems with three or more components put in parallel.

In order to attach a precise meaning to conditional axioms of the above form we need some insight in the notion of the alphabet $\alpha(X)$ of process X. We assume that X has been specified by means of a recursive specification with guarded recursion. Then one can effectively find sets

$$\alpha(\pi_1(X)) \subseteq \alpha(\pi_2(X)) \subseteq \alpha(\pi_3(X)) \subseteq \cdots \subseteq \alpha(X)$$
$$\beta(X) = \beta_1(X) \supseteq \beta_2(X) \supseteq \beta_3(X) \supseteq \cdots \supseteq \alpha(X)$$

(see 2.14, 3.5).

In general $\bigcup_{i=1}^{\infty} \alpha(\pi_i(X)) = \alpha(X)$ but $\bigcap_{i=1}^{\infty} \beta_i(X) = \alpha(X)$ need not hold (this is connected with the fact that on the basis of a given recursive specification of X the alphabet $\alpha(X)$ cannot be effectively computed, see 2.16). In practical cases, either one finds n,m such that $\alpha(\pi_m(X)) = \beta_n(X)$ $(=\alpha(X))$, or $\beta_n(X)$ is sufficiently small to verify the condition of a conditional axiom. We call this small theory about alphabets the α/β-calculus. Though very simple, this α/β calculus is an unavoidable tool in system verification based on process algebra.

In the last three sections we describe three examples and extensively demonstrate the use of CA1-7.

Concluding remarks

i) System verifications of a related nature, but performed in a model, can be found in Sifakis [13] and Olderog [12].

ii) In 2.17, we find a rather unexpected connection between Koomen's Fair Abstraction Rule (KFAR) and determination of alphabets.

iii) The consistency of CA1-7 on top of ACP_τ + KFAR + RSP (as defined in Bergstra & Klop [5]) is a nontrivial issue. This issue is not touched at all, but each of the laws is checked in detail for all finite processes.

iv) Future research in this area should involve the analysis of increasingly complicated case studies. We firmly believe that process algebra cannot be developed without extensive experimentation on case studies (examples). If this theory is brought to application, each particular application will again be a case study (at least from the point of view of the theorist), but with the special property that it is sufficiently complex or important to draw the attention of people with practical purposes. If the theorist considers the study of examples to be a part of theory, then neither the absence nor the presence of proper applications of a given body of theory will come as a surprise.

Table of contents.

1. Algebra of communicating processes with silent steps

The axiomatic framework in which we present this document is ACP$_\tau$, the algebra of communicating processes with silent steps, as described in Bergstra & Klop [4]. In this section, we give a brief review of ACP$_\tau$.

1.1 Signature.

Sorts:	A	(a finite set of atoms)
	P	(the set of processes; we have $A \subseteq P$ and elements of P are terms over A)
Functions:	$+ : P \times P \to P$	(alternative composition or sum)
	$\cdot : P \times P \to P$	(sequential composition or product)
	$\| : P \times P \to P$	(parallel composition or merge)
	$\mathbin{\|\!\!\lfloor} : P \times P \to P$	(left-merge)
	$\mid : P \times P \to P$	(communication merge; $\mid : A \times A \to A$ is given)
	$\partial_H : P \to P$	(encapsulation; $H \subseteq A$)
	$\tau_I : P \to P$	(abstraction; $I \subseteq A - \{\delta\}$)
Constants:	$\delta \in A$	(deadlock)
	$\tau \notin A$	(silent or internal action)

1.2 Axioms.

These are presented in table 1. Here $a, b \in A$, $x, y, z \in P$, $H \subseteq A$ and $I \subseteq A - \{\delta\}$.

ACP$_\tau$

$x+y=y+x$	A1	$x\tau=x$	T1
$x+(y+z)=(x+y)+z$	A2	$\tau x+x=\tau x$	T2
$x+x=x$	A3	$a(\tau x+y)=a(\tau x+y)+ax$	T3
$(x+y)z=xz+yz$	A4		
$(xy)z=x(yz)$	A5		
$x+\delta=x$	A6		
$\delta x=\delta$	A7		
$a\mid b=b\mid a$	C1		
$(a\mid b)\mid c=a\mid(b\mid c)$	C2		
$\delta\mid a=\delta$	C3		
$x\parallel y=x\lfloor\!\lfloor y+y\lfloor\!\lfloor x+x\mid y$	CM1		
$a\lfloor\!\lfloor x=ax$	CM2	$\tau\lfloor\!\lfloor x=\tau x$	TM1
$(ax)\lfloor\!\lfloor y=a(x\parallel y)$	CM3	$(\tau x)\lfloor\!\lfloor y=\tau(x\parallel y)$	TM2
$(x+y)\lfloor\!\lfloor z=x\lfloor\!\lfloor z+y\lfloor\!\lfloor z$	CM4	$\tau\mid x=\delta$	TC1
$(ax)\mid b=(a\mid b)x$	CM5	$x\mid\tau=\delta$	TC2
$a\mid(bx)=(a\mid b)x$	CM6	$(\tau x)\mid y=x\mid y$	TC3
$(ax)\mid(by)=(a\mid b)(x\parallel y)$	CM7	$x\mid(\tau y)=x\mid y$	TC4
$(x+y)\mid z=x\mid z+y\mid z$	CM8		
$x\mid(y+z)=x\mid y+x\mid z$	CM9		
		$\partial_H(\tau)=\tau$	DT
		$\tau_I(\tau)=\tau$	TI1
$\partial_H(a)=a$ if $a\notin H$	D1	$\tau_I(a)=a$ if $a\notin I$	TI2
$\partial_H(a)=\delta$ if $a\in H$	D2	$\tau_I(a)=\tau$ if $a\in I$	TI3
$\partial_H(x+y)=\partial_H(x)+\partial_H(y)$	D3	$\tau_I(x+y)=\tau_I(x)+\tau_I(y)$	TI4
$\partial_H(xy)=\partial_H(x)\cdot\partial_H(y)$	D4	$\tau_I(xy)=\tau_I(x)\cdot\tau_I(y)$	TI5

Table 1.

1.3 Concurrency.

We will also assume that the following Axioms of Standard Concurrency hold. These are proved for all finite closed ACP$_\tau$-terms in Bergstra & Klop [4].

1.	$(x\lfloor\!\lfloor y)\lfloor\!\lfloor z=x\lfloor\!\lfloor(y\parallel z)$
2.	$(x\mid ay)\lfloor\!\lfloor z=x\mid(ay\lfloor\!\lfloor z)$
3.	$x\mid y=y\mid x$
4.	$x\parallel y=y\parallel x$
5.	$x\mid(y\mid z)=(x\mid y)\mid z$
6.	$x\parallel(y\parallel z)=(x\parallel y)\parallel z$

Table 2.

2. Alphabets

2.1 Set of alphabets.

Define $\mathcal{Q}=Pow(A-\{\delta\})$, the set of all subsets of $A-\{\delta\}$. $\mathcal{Q}$ is finite, and partially ordered by inclusion, with minimal element $\varnothing$ and maximal element $A-\{\delta\}$.

2.2 Definition. We define the *alphabet* function α: $P \rightarrow \mathcal{Q}$ inductively in table 3.

1.1	$\alpha(\delta)=\varnothing$
1.2	$\alpha(a)=\{a\}$ if $a \in A-\{\delta\}$
1.3	$\alpha(\tau)=\varnothing$
2.1	$\alpha(\delta x)=\varnothing$
2.2	$\alpha(ax)=\{a\}\cup\alpha(x)$ if $a \in A-\{\delta\}$
2.3	$\alpha(\tau x)=\alpha(x)$
3.	$\alpha(x+y)=\alpha(x)\cup\alpha(y)$

Table 3.

2.3 Notes: 1. We have to check that

$$x=y \Rightarrow \alpha(x)=\alpha(y),$$

otherwise this definition is not correct. This is not hard to do.

2. Because of rules A6 and T1, we have to require that $\alpha(\delta)=\alpha(\tau)=\varnothing$, so we count only atoms different from δ and τ. This reaffirms the special status of these atoms.

2.4 Infinite processes.

Next we want to define α on infinite processes. Before we can do that, we need some additional definitions. From now on, we look at terms containing free variables (denoted by X,Y,Z, etc.). First we need the notion of guardedness.
Note: this is a more restricted version of guardedness than in Bergstra & Klop [3].

2.5 Definition.

Let t be a term containing a variable X. We say an occurrence of X in t is *guarded* if it is preceded by an atom, which will not be abstracted by some τ_I; to be precise: an occurrence of X in t is *guarded* if t has a subterm of the form aM, where $a \in A$ (so $a \notin \tau$!), and this X occurs in M and is not in the scope of an operator τ_I with $a \in I$.

2.6 Examples.

Let t be the term

$$aX + \tau X + a \mathbin{⌊\!⌊} Y + \tau_A(a+bZ) + a(X \| \tau_A(Y)).$$

In t, the first and third occurrence of X are guarded, the second is unguarded; the occurrences of Y are unguarded, and the occurrence of Z is unguarded.

2.7 Recursive specifications.

Let $<\vec{X},\vec{E}>$ be a recursive specification, i.e. a system of equations:

$$\begin{cases} X_1 = T_1(X_1,...,X_n) \\ \cdot \\ \cdot \\ \cdot \\ X_n = T_n(X_1,...,X_n). \end{cases}$$

Here $n \in \mathbb{N}$, and the $T_i(X_1,...,X_n)$ $(1 \leqslant i \leqslant n)$ are terms with free variables some (or all) of $X_1,...,X_n$.
We will look at *expansions* of the X_i $(1 \leqslant i \leqslant n)$. An expansion is formally defined by the following three properties (here s,t,u are terms with variables from $X_1,\ldots,X_n$):

1. substitution: if we obtain t from s by substituting $T_i(X_1,...,X_n)$ for an occurrence of X_i in s,

then t is an expansion of s;

2. reflexivity: t is an expansion of t;
3. transitivity: if t is an expansion of s, and u is an expansion of t, then u is an expansion of s.

2.8 Definition.

Let $<\vec{X},\vec{E}>$ be a recursive specification as in 2.7. We say $<\vec{X},\vec{E}>$ is *weakly guarded* if for all $i \leqslant n$ and for all expansions S_i of X_i, each occurrence of X_i in S_i is guarded.

We want to define the alphabet for all weakly guarded recursive specifications. The next example shows that there are some problems to be solved, before we can give the definition.

2.9 Example.

Let $<\vec{X},\vec{E}>$ be given by:

$$\begin{cases} X = iX \\ Y = \tau_{\{i\}}(X). \end{cases}$$

Intuitively, the alphabet of these processes is clear, namely $\alpha(X) = \{i\}$ and $\alpha(Y) = \varnothing$.
It becomes more difficult if we add an equation $Z = Y.a$, for, do we have $a \in \alpha(Z)$ or not? (We will see later that we must have $a \notin \alpha(Z)$.) The problem seems to be that, although each recursion is guarded, these guards may disappear when we later do abstraction.
We can solve this problem if we take care that abstraction is always the last step in our specification (i.e. if X_i is defined using abstraction, then X_i does not occur in $T_1,...,T_{i-1}, T_{i+1},...,T_n$). We call that *delayed abstraction,* so e.g. the system $<X,Y>$ above has delayed abstraction, but $<X,Y,Z>$ does not.
We can delay abstraction by adding an extra atom to A, as is done in the proof of the following theorem.

2.10 Theorem. *Let $<\vec{X},\vec{E}>$ be a recursive specification such that the communication merge $|$ does not occur in equations $\vec{E}$. Then $<\vec{X},\vec{E}>$ is equivalent to a specification with delayed abstraction.*

Proof. Add a new atom t to A, and extend the communication function by defining

$$t \,|\, a = \delta \quad (\text{for all } a \in A).$$

If x is an ACP$_\tau$-term, let x^t be the term that results if we replace all occurrences of τ by t, and all occurrences of τ_I by t_I. Here t_I is an operator defined by:

$t_I(t) = t$
$t_I(a) = a$ if $a \notin I$
$t_I(a) = t$ if $a \in I$
$t_I(x+y) = t_I(x)+t_I(y)$
$t_I(xy) = t_I(x)t_I(y)$

Table 4.
(Compare this with axioms TI1-5).

Let $<\vec{X},\vec{E}>$ be the specification

$$\begin{cases} X_1 = T_1(X_1,...,X_n) \\ \vdots \\ X_n = T_n(X_1,...,X_n). \end{cases}$$

Define a new specification by:

$$\begin{cases} X_1' = (T_1(X_1', \ldots, X_n'))^t \\ \vdots \\ X_n' = (T_n(X_1', \ldots, X_n'))^t \\ Y_1 = \tau_{\{t\}}(X_1') \\ \vdots \\ Y_n = \tau_{\{t\}}X_n') \end{cases}$$

Then we claim $X_1 = Y_1, \ldots, X_n = Y_n$. This follows from proposition 3.6 in Bergstra & Klop [4], and using the Recursive Specification Principle (RSP)

$$\boxed{\text{(RSP)} \quad \frac{<\vec{X},\vec{E}> \; <\vec{Y},\vec{E}>}{\vec{X} = \vec{Y}}}$$

which says that if X and Y satisfy the same equation, then $X = Y$ (see Bergstra & Klop [5]). It is obvious that the second specification above has delayed abstraction, so the proof is done.

2.11 Example

We consider again the specification of 2.9

$$\begin{cases} X = iX \\ Y = \tau_{\{i\}}(X) \\ Z = Y \cdot a \end{cases}$$

This specification is equivalent to the following specification with delayed abstraction:

$$\begin{cases} X = iX \\ Y' = (\tau_{\{i\}}(X))^t = t_{\{i\}}(X) \\ Z' = Y' \cdot a \\ Y = \tau_{\{t\}}(Y') \\ Z = \tau_{\{t\}}(Z') \end{cases}$$

2.12 Note. In theorem 2.10, we can allow the communication merge to occur in $\vec{E}$, if we first make terms saturated (see Bergstra & Klop, [4], 3.3-3.6).

Now we can finally define alphabets for a class of infinite processes.

2.13 Definition. Let $<\vec{X},\vec{E}>$ be a weakly guarded recursive specification. We delay abstraction as in 2.10:

$$\begin{cases} X_1' = (T_1(X_1', \ldots, X_n'))^t \\ \vdots \\ X_n' = (T_n(X_1', \ldots, X_n'))^t \\ X_1 = \tau_{\{t\}}(X_1') \\ \vdots \\ X_n = \tau_{\{t\}}(X_n') \end{cases}$$

Let π_m be the *projection* function, given on ACP_τ by ($m \in \mathbb{N}$):

$\pi_m(a) = a$	$\pi_m(\tau) = \tau$
$\pi_1(ax) = a$	$\pi_m(\tau x) = \tau \cdot \pi_m(x)$
$\pi_{m+1}(ax) = a\pi_m(x)$	
$\pi_m(x+y) = \pi_m(x) + \pi_m(y)$	

Table 5

(here $a \in A \cup \{t\}$; x,y are terms over $A \cup \{t\}$). In 2.14 we will show that each $\pi_m(X_i')$ $(1 \leqslant i \leqslant n)$ can be expanded to a closed finite ACP-term. Thus we can define

$$\boxed{\begin{array}{ll} \alpha(X_i') = \bigcup_{m=1}^{\infty} \alpha(\pi_m(X_i')) & (2.13.1) \\ \alpha(X_i) = \alpha(X_i') - \{t\} & (2.13.2) \end{array}}$$

2.14 Lemma. Let $<\vec{X},\vec{E}>$ be a weakly guarded recursive specification, in which no τ or τ_I occurs. Then each $\pi_m(X_i)$ can be expanded to a closed finite ACP-term .

Proof. The proof is by induction on m. It is not hard to see, that it is enough to show, that each X_i can be expanded to a term S_i in which all occurrences of variables are guarded (the general result then follows by iteration).

We obtain such a S_i as follows.
We define for each $m \in \mathbb{N}$ the m^{th} *expansion* of X_i, $\mathcal{E}_i^m$, as follows:

$$\mathcal{E}_i^1 = T_i(X_1, \ldots, X_n)$$

$$\mathcal{E}_i^2 = T_i(T_1(\vec{X}),...,T_n(\vec{X}))$$

$$\mathcal{E}_i^m = T_i(T_1(...(T_1(\vec{X}),...,T_n(\vec{X})),...,...),...,T_n(...,...,...))$$

We put $S_i = \mathcal{E}_i^n$. Then, if we look at a variable occurrence in the resulting S_i, it resulted from $(n+1)$ successive substitutions. Since there are only n variables, at least one variable, say X_j, must have occurred twice in this sequence. But since recursion is guarded, the second occurrence of X_j must be guarded. The variable in S_i we started from results from an expansion of X_j, so is also guarded.

2.15 Note: in definition 2.13, the partial unions $\bigcup_{m=1}^{N} \alpha(\pi_m(X_i'))$ form an increasing sequence (with respect to $\subseteq$) as $N \to \infty$, in the finite set $\mathcal{Q}$, so the sequence must be eventually constant, and the limit will always exist.
However, in general it is *undecidable,* to which set in $\mathcal{Q}$ $\alpha(X_i)$ is equal. This is illustrated in the following example.

2.16 Example.

Let K be a r.e. but not recursive subset of $\mathbb{N}$. In Bergstra & Klop [3] a recursive specification $<\vec{X},\vec{E}>$ is given (depending on n) such that in the initial algebra the following hold:

$$X_1(n) = b^{\omega} \text{ if } n \notin K$$

$$X_1(n) = b^k \cdot stop \text{ if } n \in K \quad (\text{for some } k \in \mathbb{N}).$$

(here b, *stop* are atoms).
Thus we have $\alpha(X_1(n)) = \{b\}$ if $n \notin K$ and $\alpha(X_1(n)) = \{b, stop\}$ if $n \in K$.
Since K is not recursive, determining whether $n \in K$, for a given n, is undecidable, so determining $\alpha(X_1)$ is undecidable.

2.17 Example.

Again consider specification $<X,Y,Z>$ from 2.9. We defined the delayed abstraction in 2.11. Here we will determine the alphabets. First, we consider X. Let $m \in \mathbb{N}$, then $X = iX = iiX = \cdots = \underbrace{ii\cdots i}_{m\times}X$, so $\pi_m(X) = ii...i$ and $\alpha(\pi_m(X)) = \{i\}$ by 2.2. Thus by 2.13 we have $\alpha(X) = \{i\}$. Next,

$$Y' = (\tau_{\{i\}}(X))^t = t_{\{i\}}(X) = t_{\{i\}}(iX) =$$

$$t_{\{i\}}(i)t_{\{i\}}(X) = tY'.$$

Then we get $\alpha(Y') = \{t\}$ as above. But now, if $m \in \mathbb{N}$, then

$$\pi_m(Z') = \pi_m(Y'a) = \pi_m(\underbrace{tt\cdots t}_{m\times}Y'a) = tt...t,$$

whence $\alpha(Z')=\{t\}$. Finally, by 2.13 we get

$$\alpha(Y) = \{t\}-\{t\} = \varnothing \text{ and } \alpha(Z) = \{t\}-\{t\} = \varnothing.$$

We can motivate these results in a different way, if we use Koomen's Fair Abstraction Rule (KFAR, see Bergstra & Klop [5]):

$$\text{(KFAR)} \quad \frac{\forall n \in \mathbb{Z}_k \quad X_n = i_n.X_{n+1} + Y_n \quad (i_n \in I)}{\tau_I(X_n) = \tau.\tau_I(Y_0 + ... + Y_{k-1})}$$

Here $\mathbb{Z}_k = \{0,...,k-1\}$ and addition in subscripts works modulo k. Here we will use the following (easier to understand) consequence of KFAR:

$$(\text{KFAR},k=1) \quad \frac{X = iX + Y}{\tau_{\{i\}}(X) = \tau.\tau_{\{i\}}(Y)}$$

This expresses the fact that, due to some fairness mechanism, i resists being performed infinitely many times consecutively. Here, we have

$$X = iX = iX + \delta,$$

so applying KFAR yields

$$Y = \tau_{\{i\}}(X) = \tau\delta.$$

Then $Z = Ya = \tau\delta a = \tau\delta$, and we see again that $\alpha(Y) = \alpha(Z) = \varnothing$.

3. α/β-calculus

3.1 Definition 2.13.1 gives a sequence of subsets of $\alpha(X_i)$, which will converge to $\alpha(X_i)$. However, as 2.16 illustrated, finding $\alpha(X_i)$ itself can sometimes be very difficult. Luckily, in applications it is often sufficient to have a superset of $\alpha(X_i)$, which is not too big. With such a superset, we can even determine $\alpha(X_i)$ in many cases. For this reason, we define $\beta(X_i)$ in 3.4. First we need theorem 3.2. A piece of notation: if $B,C \subseteq A \cup \{\tau\}$, we define $B \mid C = \{b \mid c : b \in B, c \in C\} - \{\delta\}$ (We leave out δ, so that $B \mid C \in \mathcal{Q}$).

3.2 Theorem: *for all closed finite $ACP_\tau-$ terms x,y:*

1. $\alpha(xy) \subseteq \alpha(x) \cup \alpha(y)$
2. $\alpha(x \| y) = \alpha(x) \cup \alpha(y) \cup \alpha(x) \mid \alpha(y)$
3. $\alpha(x \mathbin{⌊\!⌊} y) \subseteq \alpha(x \| y)$
4. $\alpha(x \mid y) \subseteq \alpha(x \| y)$
5. $\alpha(\partial_H(x)) \subseteq \alpha(x) - H$
6. $\alpha(\tau_I(x)) = \alpha(x) - I$.

Proof.

1. by induction on x. We have seven cases (as in 2.2):

1.1 $x = \delta$. $\alpha(\delta y) = \varnothing \subseteq \alpha(\delta) \cup \alpha(y)$

1.2 $x = a \neq \delta$. $\alpha(ay) = \{a\} \cup \alpha(y) = \alpha(a) \cup \alpha(y)$

1.3 $x = \tau$. $\alpha(\tau y) = \alpha(y) = \alpha(\tau) \cup \alpha(y)$

1.4 $x = \delta z$. As 1.1.

1.5 $x = az, a \neq \delta$. $\alpha(azy) = \{a\} \cup \alpha(zy) \subseteq \{a\} \cup \alpha(z) \cup \alpha(y) = \alpha(az) \cup \alpha(y)$.

1.6 $x = \tau z$. $\alpha(\tau zy) = \alpha(zy) \subseteq \alpha(z) \cup \alpha(y) = \alpha(\tau z) \cup \alpha(y)$

1.7 $x = z + w$. $\alpha((z+w)y) = \alpha(zy + wy) = \alpha(zy) \cup \alpha(wy) \subseteq \alpha(z) \cup \alpha(y) \cup \alpha(w) \cup \alpha(y) = \alpha(z+w) \cup \alpha(y)$.

2. This is more complicated. We do simultaneous induction on x and y, and write

$$x = \sum_{i=1}^{I} a_i x_i' + \sum_{j=1}^{J} \tau x_j'' + \sum_{k=1}^{K} b_k + (\tau+\delta)$$

($a_i, b_k \in A - \{\delta\}$, the part between brackets may or may not occur, $I, J, K \geqslant 0$),

$$Y = \sum_{\ell=1}^{L} c_\ell y_\ell' + \sum_{m=1}^{M} \tau y_m'' + \sum_{n=1}^{N} d_n + (\tau+\delta)$$

$$(c_\ell, d_n \in A - \{\delta\}, L, M, N \geqslant 0).$$

By induction hypothesis we can assume 2 holds for all terms

$$x \| y_\ell',\ x \| y_m'',\ x_i' \| y,\ x_i' \| y_\ell',\ x_i' \| y_m'',$$
$$x_j'' \| y,\ x_j'' \| y_\ell',\ x_j'' \| y_m''.$$

To expand $x \| y$, we use rules CM1-CM9; TM1-2 and TC1-4:

$$\begin{aligned}
x \| y = {} & \sum_{i=1}^{I} a_i(x_i' \| y) + \sum_{j=1}^{J} \tau(x_j'' \| y) + \sum_{k=1}^{K} b_k y + (\tau y) + \\
& + \sum_{\ell=1}^{L} c_\ell(x \| y_\ell') + \sum_{m=1}^{M} \tau(x \| y_m'') + \sum_{n=1}^{N} d_n x + (\tau x) + \\
& + \sum_{i=1}^{I}\sum_{\ell=1}^{L} (a_i \mid c_\ell)(x_i' \| y_\ell') + \boxed{\sum_{i=1}^{I}\sum_{m=1}^{M} (a_i x_i') | (\tau y_m'')} + \sum_{i=1}^{I}\sum_{n=1}^{N} (a_i \mid d_n) x_i' + \\
& + \boxed{\sum_{j=1}^{J}\sum_{\ell=1}^{L} (\tau x_j'') | (c_\ell y_\ell') + \sum_{j=1}^{J}\sum_{m=1}^{M} (\tau x_j'') | (\tau y_m'') + \sum_{j=1}^{J}\sum_{n=1}^{N} (\tau x_j'') | d_n} + \\
& + \sum_{k=1}^{K}\sum_{\ell=1}^{L} (b_k | c_\ell) y_\ell' + \boxed{\sum_{k=1}^{K}\sum_{m=1}^{M} b_k | (\tau y_m'')} + \sum_{k=1}^{K}\sum_{n=1}^{N} b_k | d_n + \delta.
\end{aligned}$$

Now the five enclosed terms can be skipped, since they are summands of other terms (for instance, $\sum_{K \times M} b_k \mid (\tau y_m'') = \sum_{K \times M} b_k \mid y_m''$ is a summand of $\sum_M \tau\,(x \| y_m'')$, see Bergstra & Klop [4], 3.6). Now we use definition 2.2, and obtain the following (at some places we have to write subsets, because some communications might be δ).

$$\begin{aligned}
\alpha(x \| y) = {} & [\bigcup_{i=1}^{I} (\{a_i\} \cup \alpha(x_i' \| y)) \cup \bigcup_{j=1}^{J} \alpha(x_j'' \| y) \cup \bigcup_{k=1}^{K} \{b_k\} \cup \alpha(y) \cup \\
& \cup \bigcup_{\ell=1}^{L} (\{c_\ell\} \cup \alpha(x \| y_\ell')) \cup \bigcup_{m=1}^{M} \alpha(x \| y_m'') \cup \bigcup_{n=1}^{N} \{d_n\} \cup \alpha(x) \cup \\
& \cup \bigcup_{\substack{(i,\ell) \in a \\ \text{subset of } K \times L}} (\{a_i \mid c_\ell\} \cup \alpha(x_i' \| y_\ell')) \cup \bigcup_{\substack{(i,n) \in a \\ \text{subset of } I \times N}} (\{a_i \mid d_n\} \cup \alpha(x_i')) \cup \\
& \cup \bigcup_{\substack{(k,\ell) \in a \\ \text{subset of } K \times L}} (\{b_k \mid c_\ell\} \cup \alpha(y_\ell')) \cup \bigcup_{(k,n) \in K \times N} \{b_k \mid d_n\}] - \{\delta\}.
\end{aligned}$$

Next we use the induction hypothesis.

$$\begin{aligned}
\alpha(x \| y) = {} & [\bigcup_{i=1}^{I} (\{a_i\} \cup \alpha(x_i') \cup \alpha(x_i') \mid \alpha(y)) \cup \alpha(y) \cup \\
& \bigcup_{j=1}^{J} (\alpha(x_j'') \cup \alpha(x_j'') \mid \alpha(y)) \cup \alpha(y) \cup \bigcup_{k=1}^{K} \{b_k\} \cup \alpha(y) \cup \\
& \bigcup_{\ell=1}^{L} (\{c_\ell\} \cup \alpha(y_\ell') \cup \alpha(x) \mid \alpha(y_\ell')) \cup \alpha(x) \cup
\end{aligned}$$

$$\bigcup_{m=1}^{M}(\alpha(y_m'')\cup\alpha(x)\mid\alpha(y_m''))\cup\alpha(x)\cup\bigcup_{n=1}^{N}\{d_n\}\cup\alpha(x)\cup$$

$$\bigcup_{\subseteq I\times L}(\{a_i\mid c_\ell\}\cup\alpha(x_i')\cup\alpha(y_\ell')\cup\alpha(x_i')\mid\alpha(y_\ell'))\cup$$

$$\bigcup_{\subseteq I\times N}(\{a_i\mid d_n\}\cup\alpha(x_i'))\cup\bigcup_{\subseteq K\times L}(\{b_k\mid c_\ell\}\cup\alpha(y_\ell'))\cup\bigcup_{K\times N}\{b_k\mid d_n\}]-\{\delta\} =$$

$$= [\bigcup_{i=1}^{I}(\{a_i\}\cup\alpha(x_i'))\cup\bigcup_{j=1}^{J}\alpha(x_j'')\cup\bigcup_{k=1}^{K}\{b_k\}\cup\alpha(x)\cup$$

$$\bigcup_{\ell=1}^{L}(\{c_\ell\}\cup\alpha(y_\ell'))\cup\bigcup_{m=1}^{M}\alpha(y_m'')\cup\bigcup_{n=1}^{N}\{d_n\}\cup\alpha(y)\cup$$

$$\bigcup_{i=1}^{I}(\alpha(x_i')\mid\alpha(\sum_{\ell=1}^{L}c_\ell y_\ell' + \sum_{m=1}^{M}\tau y_m'' + \sum_{n=1}^{N}d_n + (\tau+\delta)))\cup$$

$$\bigcup_{j=1}^{J}(\alpha(x_j'')\mid\alpha(\sum_{\ell=1}^{L}c_\ell y_\ell' + \sum_{m=1}^{M}\tau y_m'' + \sum_{n=1}^{N}d_n + (\tau+\delta)))\cup$$

$$\bigcup_{\ell=1}^{L}\alpha(x)\mid\alpha(y_\ell')\cup\bigcup_{m=1}^{M}\alpha(x)\mid\alpha(y_m'')\cup$$

$$\bigcup_{I\times L}\{a_i\mid c_\ell\}\cup\bigcup_{I\times N}\{a_i\mid d_n\}\cup\bigcup_{K\times L}\{b_k\mid c_\ell\}\cup\bigcup_{K\times N}\{b_k\mid d_n\}]-\{\delta\}$$

(for $\alpha(x_i')\mid\alpha(y_\ell')\subseteq\alpha(x)\mid\alpha(y_\ell')$ for all $i\in I,\ell\in L$) =

$$= [\alpha(x)\cup\alpha(y)\cup\bigcup_{\ell=1}^{L}\alpha(x)\mid\alpha(y_\ell')\cup\bigcup_{m=1}^{M}\alpha(x)\mid\alpha(y_m'')\cup$$

$$\bigcup_{I\times L}\alpha(x_i')\mid\{c_\ell\}\cup\bigcup_{I\times N}\alpha(x_i')\mid\{d_n\}\cup$$

$$\bigcup_{J\times L}\alpha(x_j'')\mid\{c_\ell\}\cup\bigcup_{J\times N}\alpha(x_j'')\mid\{d_n\}\cup$$

$$\bigcup_{I\times L}\{a_i\mid c_\ell\}\cup\bigcup_{K\times L}\{b_k\mid c_\ell\}\cup\bigcup_{I\times N}\{a_i\mid d_n\}\cup\bigcup_{K\times N}\{b_k\mid d_n\}]-\{\delta\}$$

= (take the boxed areas together)

$$[\alpha(x)\cup\alpha(y)\cup\bigcup_{L}\alpha(x)\mid\alpha(y_\ell')\cup\bigcup_{M}\alpha(x)\mid\alpha(y_m'')\cup$$

$$\bigcup_{L}\alpha(x)\mid\{c_\ell\}\cup\bigcup_{N}\alpha(x)\mid\{d_n\}]-\{\delta\} =$$

$$= \alpha(x)\cup\alpha(y)\cup\alpha(x)\mid\alpha(y).$$

This finishes the proof of 3.2.2.

3,4. These follow from 2.2.3 with rule CM1

5. By induction on x. We have 8 cases:

5.1 $x=a\in H$. $\alpha(\partial_H(a)) = \alpha(\delta) = \varnothing = \{a\}-H = \alpha(a)-H$

5.2 $x=a\notin H$. $\alpha(\partial_H(a))=\alpha(a)=\{a\}=\{a\}-H=\alpha(a)-H$

5.3 $x=\tau$. $\alpha(\partial_H(\tau))=\alpha(\tau)=\varnothing=\alpha(\tau)-H$

5.4 $x=\delta y$. As 5.1.

5.5 $x=ay,\ a\in H\ .\ \alpha(\partial_H(ay))=\alpha(\delta.\partial_H(y))=a(\delta)=\varnothing\subseteq\alpha(ay)-H$

5.6 $x=ay,\ a\notin H,\ a\neq\delta.\ \alpha(\partial_H(ay))=\alpha(a.\partial_H(y))=\{a\}\cup\alpha(\partial_H(y))\subseteq\{a\}\cup\alpha(y)-H\ =\ \alpha(ay)-H.$

5.7 $x=\tau y\ .\ \alpha(\partial_H(\tau y))=\alpha(\tau.\partial_H(y))=\alpha(\partial_H(y))\subseteq\alpha(y)-\ H=\alpha(\tau y)-H.$

5.8 $x=y+z\ .\ \alpha(\partial_H(y+z))=\alpha(\partial_H(y)+\partial_H(z))=\alpha(\partial_H(y))\cup\alpha(\partial_H(z))\subseteq$
$\subseteq(\alpha(y)-H)\cup(\alpha(z)-H)=(\alpha(y)\cup\alpha(z))-H=\alpha(y+z)-H.$

6. By induction on x. We have 9 cases:

6.1 $x=\delta.\ \alpha(\tau_I(\delta))=\alpha(\delta)=\varnothing=\alpha(\delta)-I.$

6.2 $x=a\in I\ .\ \alpha(\tau_I(a))=\alpha(\tau)=\varnothing=\{a\}-I=\alpha(a)-I.$

6.3 $x=a\notin I,\ a\neq\delta.\ \alpha(\tau_I(a))=\alpha(a)=\{a\}=\{a\}-I=\alpha(a)-I.$

6.4 $x=\tau.\ \alpha(\tau_I(\tau))=\alpha(\tau)=\varnothing=\alpha(\tau)-I.$

6.5 $x=\delta y\ .\ \alpha(\tau_I(\delta y))=\alpha(\delta)=\varnothing=\alpha(\delta y)-I.$

6.6 $x=ay,\ a\in I\ .\ \alpha(\tau_I(ay))=\alpha(\tau.\tau_I(y))=\alpha(\tau_I(y))=\alpha(y)-I=\ (\alpha(y)\cup\{a\})-I=\alpha(ay)-I.$

6.7 $x=ay,\ a\notin I,\ a\neq\delta.\ \alpha(\tau_I(ay))=\alpha(a.\tau_I(y))=\{a\}\cup\alpha(\tau_I(y))=$
$\{a\}\cup(\alpha(y)-I)=(\{a\}\cup\alpha(y))-I=\alpha(ay)-I.$

6.8 $x=\tau y\ .\ \alpha(\tau_I(\tau y))=\alpha(\tau.\tau_I(y))=\ \alpha(\tau_I(y))=\alpha(y)-I=\alpha(\tau y)-I.$

6.9 $x=y+z\ .\ \alpha(\tau_I(y+z))=\alpha(\tau_I(y)+\tau_I(z))=\alpha(\tau_I(y))\cup\alpha(\tau_I(z))=$
$(\alpha(y)-I)\cup(\alpha(z)-I)=(\alpha(y)\cup\alpha(z))-I=\alpha(y+z)-I.$

This finishes the proof of Theorem 3.2.

3.3 For each ACP$_\tau$-term involving free variables, we want to define a corresponding set-term, involving the alphabets of these variables. First two remarks:

1. Note that formulas 2.2.1-1-3 hold for infinite processes as well. For instance, if $\alpha(X)$ and $\alpha(Y)$ are given by 2.13.1, then

$$\alpha(X+Y)\ =\ \bigcup_{m=1}^{\infty}\alpha(\pi_m(X+Y)))\ =\ \bigcup_{m=1}^{\infty}\alpha(\pi_m(X)+\pi_m(Y))\ =$$
$$=\ \bigcup_{m=1}^{\infty}(\alpha(\pi_m(X))\cup\alpha(\pi_m(Y)))\ =\ \alpha(X)\cup\alpha(Y).$$

2. Assume that formulas 3.2.1-6 hold for infinite processes as well.

Now let $T\ =\ T(X_1,\ldots,X_n)$ be a ACP$_\tau$-term with free variables among $X_1,\ldots,X_n$. We apply rules 2.2 and 3.2 to $\alpha(T)$. We work from the outside in, and go on until we only have unknowns $\alpha(X_j)$ left, so α is not applied to any composite term. We obtain

$$\alpha(T)\ \subseteq\ T^*(\alpha(X_1),\ldots,\alpha(X_n)).$$

Here T^* is a term over the following signature:

Sort: $\mathcal{Q}$ (subsets of $A-\{\delta\}$)

Functions: $\cup\ :\mathcal{Q}\times\mathcal{Q}\rightarrow\mathcal{Q}$ (union)

$-H:\mathcal{Q}\times\mathcal{Q}\rightarrow\mathcal{Q}$ (leave out elements of $H\in\mathcal{Q}$)

$|:\mathcal{Q}\times\mathcal{Q}\rightarrow\mathcal{Q}$ (communication, see 3.1)

Constants: $\{a\}\in\mathcal{Q}$ (for each $a\in A-\{\delta\}$)

Example: if $T=a.\tau_I(X_4)(X_1\|X_2+\partial_H(X_3))$, then $T^*=\{a\}\cup\alpha(X_1)\cup\alpha(X_2)\cup\ (\alpha(X_3)-H)\cup$
$\cup(\alpha(X_4)-I)\cup\ \alpha(X_1)\,|\,\alpha(X_2)$

3.4 Definition.

Let $<\vec{X},\vec{E}>$ be a weakly guarded recursive specification, and $\vec{X}\ =\ X_1,\ldots,X_n,\ 1\leqslant i\leqslant n.$ We define $\beta(X_i)$ to be the least fixed point of the equation (over $\mathcal{Q}$):

$$\beta(X_i) = T_i^*(\beta(X_1), \ldots, \beta(X_n)).$$

Note that this least fixed point will always exist, since terms T^* over $(\mathcal{Q}, \cup, -H, |, \{a\})$ are *monotonic*, i.e. $B_i \subseteq B_i'$ implies

$$T^*(B_1, \ldots, B_i, \ldots, B_n) \subseteq T^*(B_1,\ldots,B_i', \ldots, B_n).$$

Thus, $\beta(X_i)$ is the limit of successive approximations

$$T_i^*(\varnothing, \ldots, \varnothing),$$
$$T_i^*(T_1^*(\varnothing, \ldots, \varnothing), \ldots, T_n^*(\varnothing, \ldots, \varnothing)),$$
$$\ldots, etc.$$

3.5 Theorem. *Let* $<\vec{X},\vec{E}>$ *be a weakly guarded recursive specification given by*

$$\begin{cases} X_1 = T_1(X_1, \ldots, X_n) \\ \vdots \\ X_n = T_n(X_1,\ldots,X_n) \end{cases}$$

Let $1 \leqslant i \leqslant n$. *Then* $\alpha(X_i) \subseteq \beta(X_i)$.

Proof. Let $\mathcal{E}_i^m$ be the m-th expansion of X_i as defined in 2.14. We denote the $\beta(X_i)$, belonging to specification

$$<\vec{X},<\mathcal{E}_i^m \mid 1 \leqslant i \leqslant n>>, \text{ by } \beta_m(X_i) \text{ (so } \beta(X_i) = \beta_1(X_i)).$$

Claim: $\beta_m(X_i) \subseteq \beta(X_i)$. (for all $m \in \mathbb{N}$).
Proof of claim: Take $m = 2$. Now

$$(\mathcal{E}_i^2)^*(\beta(X_1),\ldots,\beta(X_n)) =$$
$$T_i^*(T_1^*(\beta(X_1),\ldots,\beta(X_n)),\ldots,T_n^*(\beta(X_1),\ldots,\beta(X_n))) =$$
$$= T_i^*(\beta(X_1),\ldots,\beta(X_n)) = \beta(X_i).$$

Therefore, $\beta(X_i)$ is a fixed point of the equation

$$B_i = (\mathcal{E}_i^2)^*(B_1, \ldots, B_n)$$

Since $\beta_2(X_i)$ is the *least* fixed point, we have $\beta_2(X_i) \subseteq \beta(X_i)$. The general case follows by iteration.

Now we delay abstraction in $<\vec{X},\vec{E}>$ as in 2.10, and obtain:

$$\begin{cases} X_1' = T_1^t(X_1',\ldots,X_n') \\ \vdots \\ X_n' = T_n^t(X_1',\ldots,X_n') \\ X_1 = \tau_{\{t\}}(X_1') \\ \quad\vdots \\ X_n = \tau_{\{t\}}(X_n') \end{cases}$$

First we show $\alpha(X_i') \subseteq \beta(X_i')$. Let $m \in \mathbb{N}$. By 2.14, there is an $k \in \mathbb{N}$ such that $\pi_m(\mathcal{E}_i^k)$ is an ACP-term without variables (all the X_j' are pushed down below level m). Thus

$$\alpha(\pi_m(X_i')) = \alpha(\pi_m(\mathcal{E}_i^k)) = \alpha(\pi_m(\mathcal{E}_1^k(X_1',\ldots,X_n'))) =$$

$$= \alpha(\pi_m(\mathcal{E}_i^k(\delta, \ldots, \delta))) \subseteq \alpha(\mathcal{E}_i^k(\delta, \ldots, \delta)) \subseteq \qquad \text{(by 3.3.1-2)}$$

$$\subseteq \mathcal{E}_i^{k*}(\varnothing, \ldots, \varnothing) \subseteq \beta_k(X_i') \subseteq \beta(X_i'),$$

and because $\alpha(X_i')$ is defined by 2.13.1, it follows that $\alpha(X_i') \subseteq \beta(X_i')$. Finally

$$\alpha(X_i) = \alpha(X_i') - \{t\} \subseteq \beta(X_i') - \{t\} = \beta(X_i)$$

and this last equality holds because $\beta(X_i)$ is defined to be the least fixed point of

$$\beta(X_i) = \tau_{\{t\}}{}^*(\beta(X_i')) = \beta(X_i') - \{t\} \qquad (3.2.6)$$

3.6 Notes. 1. It can be shown that the fixed point $\beta(X_i)$ can be reached from $\varnothing$ in at most n iterations, if we assume the Handshaking Axiom (HA, see Bergstra & Tucker [8]): $a \mid b \mid c = \delta$;
2. We cannot have in general that $\bigcap_{m=1}^{\infty} \beta_m(X_i) = \alpha(X_i)$, because that would make the determination of $\alpha(X_i)$ decidable, which contradicts 2.16.

3.7 Example. A bag is given by the recursive definition (see Bergstra & Klop [3]):
$B = \sum_{d \in D} r(d)(s(d) \| B)$.

Here D is a finite set of data, $r(d)$ means *receive* d and $s(d)$ means *send* d. The set of atoms contains $\{r(d), s(d) \mid d \in D\}$ and there is no communication (for all $a, b \in A$ we have $a \mid b = \delta$). Now

$$\alpha(\pi_2(B)) = \alpha(\sum_{d \in D} r(d)(s(d) + \sum_{e \in D} r(e)) =$$

$$= \bigcup_{d \in D}(\{r(d)\} \cup \{s(d)\} \cup \{r(d)\}) = \{r(d), s(d) \mid d \in D\}.$$

Also

$$\alpha(B) = \bigcup_{d \in D}(\{r(d)\} \cup \alpha(s(d) \| B)) =$$

$$\bigcup_{d \in D}(\{r(d)\} \cup \{s(d)\} \cup \alpha(B) \cup \{s(d)\} \mid B).$$

Therefore $\beta(B) = \{r(d), s(d) \mid d \in D\}$. Since $\alpha(\pi_2(B)) \subseteq \alpha(B) \subseteq \beta(B)$, we must have $\alpha(B) = \{r(d), s(d) \mid d \in D\}$. In this way we can calculate the alphabet of many interesting specifications. We will make extensive use of this so-called α/β-calculus in the following paragraphs.

4. Conditional axioms

We present 7 conditional axioms, which we will use in the following paragraphs. We prove these axioms for all closed finite ACP$_\tau$-terms, and postulate them for all infinite terms. See table 6.

4.1 Proof of CA1 for all closed ACP$_\tau$-terms, by simultaneous induction on x and y. Put

$$x = \sum_{i=1}^{I} a_i x_i' + \sum_{j=1}^{J} \tau x_j'' + \sum_{k=1}^{K} b_k + (\tau + \delta)$$

$$y = \sum_{\ell=1}^{L} c_\ell y_\ell' + \sum_{r=1}^{R} h_r z_r + \sum_{m=1}^{M} \tau y_m'' + \sum_{n=1}^{N} d_n + \sum_{s=1}^{S} k_s + (\tau + \delta).$$

Here $a_i, b_k \in A - \{\delta\}$; $I, J, K \geqslant 0$,

$c_\ell, d_n \in A \ (H \cup \{\delta\})$,

$h_r, k_s \in (A \cap H) - \{\delta\}; L, M, N, R, S \geqslant 0$.

Now $a_i, b_k \in \alpha(x)$ and $h_r, k_s \in \alpha(y) \cap H$, so by assumption $a_i \mid h_r$, $a_i \mid k_s$, $b_k \mid h_r$, $b_k \mid k_s \in H$.

$\dfrac{\alpha(x)\,\vert\,(\alpha(y)\cap H)\subseteq H}{\partial_H(x\,\Vert\,y)=\partial_H(x\,\Vert\,\partial_H(y))}$	CA1	$\dfrac{\alpha(x)\,\vert\,(\alpha(y)\cap I)=\varnothing}{\tau_I(x\,\Vert\,y)=\tau_I(x\,\Vert\,\tau_I(y))}$	CA2
$\dfrac{\alpha(x)\cap H=\varnothing}{\partial_H(x)=x}$	CA3	$\dfrac{\alpha(x)\cap I=\varnothing}{\tau_I(x)=x}$	CA4
$\dfrac{H=H_1\cup H_2}{\partial_H(x)=\partial_{H_1}\circ\partial_{H_2}(x)}$	CA5	$\dfrac{I=I_1\cup I_2}{\tau_I(x)=\tau_{I_1}\circ\tau_{I_2}(x)}$	CA6
$\dfrac{H\cap I=\varnothing}{\tau_I\circ\partial_H(x)=\partial_H\circ\tau_I(x)}$	CA7		

Table 6.

$$\begin{aligned}
\partial_H(x\,\Vert\,y) = {} & \sum_{i=1}^{I}\partial_H(a_i)\cdot\partial_H(x_i'\,\Vert\,y)+\sum_{j=1}^{J}\tau.\partial_H(x_j''\,\Vert\,y)+\sum_{k=1}^{K}\partial_H(b_k)\cdot\partial_H(y)\,+\\
& +\sum_{\ell=1}^{L}c_\ell\partial_H(x\,\Vert\,y_\ell')+\sum_{m=1}^{M}\tau.\partial_H(x\,\Vert\,y_m'')+\sum_{n=1}^{N}d_n.\partial_H(x)+\\
& +\sum_{I\times L}\partial_H(a_i\,\vert\,c_\ell).\partial_H(x_i'\,\Vert\,y_\ell')+\boxed{\sum_{I\times M}\partial_H((a_ix_i')\,\vert\,(\tau y_m''))}+\\
& +\sum_{I\times N}\partial_H(a_i\,\vert\,d_n)\cdot\partial_H(x_i')+\boxed{\sum_{J\times L}\partial_H((\tau x_j'')\,\vert\,(c_\ell y_\ell'))}+\\
& +\boxed{\sum_{J\times M}\partial_H((\tau x_j'')\,\vert\,(\tau y_m''))}+\boxed{\sum_{J\times N}\partial_H((\tau x_j'')\,\vert\,d_n)}+\\
& +\sum_{K\times L}\partial_H(b_k\,\vert\,c_\ell)\cdot\partial_H(y_\ell')+\boxed{\sum_{K\times M}\partial_H(b_k\,\vert\,(\tau y_m''))}+\sum_{K\times N}\partial_H(b_k\,\vert\,d_n)+(\tau+\delta)
\end{aligned}$$

As in the proof of 3.2.2, we see that we can skip the five enclosed terms. Now we apply the induction hypothesis, which is possible since $\alpha(x_i')$, $\alpha(x_j'')\subseteq\alpha(x)$ and $\alpha(y_\ell')$, $\alpha(y_m'')\subseteq\alpha(y)$.

$$\begin{aligned}
\partial_H(x\,\Vert\,y) = {} & \sum_{i=1}^{I}\partial_H(a_i)\cdot\partial_H(x_i'\,\Vert\,\partial_H(y))+\sum_{j=1}^{J}\tau.\partial_H(x_j''\,\Vert\,\partial_H(y))+\\
& +\sum_{k=1}^{K}\partial_H(b_k)\partial_H(y)+\sum_{\ell=1}^{L}c_\ell\partial_H(x\,\Vert\,\partial_H(y_\ell'))+\\
& +\sum_{m=1}^{M}\tau\partial_H(x\,\Vert\,\partial_H(y_m''))+\sum_{n=1}^{N}d_n.\partial_H(x)+\\
& +\sum_{I\times L}\partial_H(a_i\,\vert\,c_\ell)\partial_H(x_i'\,\Vert\,\partial_H(y_\ell'))+\sum_{I\times N}\partial_H(a_i\,\vert\,d_n)\partial_H(x_i')+\\
& +\sum_{K\times L}\partial_H(b_k\,\vert\,c_\ell)\cdot\partial_H(y_\ell')+\sum_{K\times N}\partial_H(b_k\,\vert\,d_n)+(\tau+\delta).
\end{aligned}$$

We use the same argument (as in 3.2.2) again to add the terms:

$$\begin{aligned}
& \sum_{I\times M}\partial_H(a_ix_i'\,\vert\,\tau\partial_H(y_m''))+\sum_{J\times L}\partial_H(\tau x_j''\,\vert\,c_\ell\partial_H(y_\ell'))+\\
& +\sum_{J\times M}\partial_H(\tau x_j''\,\vert\,\tau\partial_H(y_m''))+\sum_{J\times N}\partial_H(\tau x_j''\,\vert\,d_n)+\\
& +\sum_{K\times M}\partial_H(b_k\,\vert\,\tau\partial_H(y_m'')).
\end{aligned}$$

Then it follows that

$$\partial_H(x \| y) = \partial_H([\sum a_i x_i' + \sum \tau x_j'' + \sum b_k + (\tau+\delta)] \|$$
$$[\sum c_l \partial_H(y_l') + \sum \tau \partial_H(y_m'') + \sum d_n + (\tau+\delta)]) =$$
$$= \partial_H(x \| \partial_H(y)).$$

4.2 Proof of CA2 for all closed ACP$_\tau$-terms. This is entirely similar to 4.1. Note that now we have to have $a_i | h_r = \delta$ (if $a_i \in \alpha(x)$, $h_r \in \alpha(y) \cap I$), not just $a_i | h_r \in I$, so that $\tau_I((a_i x_i') | (h_r z_r)) = \tau_I((a_i | h_r)(x_i' \| z_r)) = \delta$, and all these terms drop out.

4.3 Proof of CA3 for all closed ACP$_\tau$-terms is by induction on x. We have 7 cases:
1. $x = \delta . \partial_H(\delta) = \delta$.
2. $x = a \neq \delta$. Since $\varnothing = \alpha(x) \cap H = \{a\} \cap H$, we have $a \notin H$. Then $\partial_H(a) = a$ (D1).
3. $x = \tau$. $\partial_H(\tau) = \tau$ (DT).
4. $x = \delta y$. $\partial_H(\delta y) = \delta = \delta y$.
5. $x = ay, a \neq \delta$. $\varnothing = \alpha(x) \cap H = (\{a\} \cup \alpha(y)) \cap H$. Thus $a \notin H$ and $\alpha(y) \cap H = \varnothing$, whence $\partial_H(ay) = \partial_H(a)\partial_H(y) = ay$.
6. $x = \tau y$. $\varnothing = \alpha(x) \cap H = \alpha(y) \cap H$, so $\partial_H(\tau y) = \tau \partial_H(y) = \tau y$.
7. $x = y + z$. $\varnothing = \alpha(x) \cap H = (\alpha(y) \cup \alpha(z)) \cap H$, so $\alpha(y) \cap H = \varnothing$ and $\alpha(z) \cap H = \varnothing$. Then $\partial_H(y+z) = \partial_H(y) + \partial_H(z) = y + z$.

4.4 Proof of CA4 for all closed ACP$_\tau$-terms is very similar to 4.3.

4.5 Proof of CA5 for all closed ACP$_\tau$-terms is by induction on x. We have 5 cases:
1. $x = a, a \notin H$. Then also $a \notin H_1, a \notin H_2$. Thus $\partial_H(a) = a = \partial_{H_1}(a) = \partial_{H_1}(\partial_{H_2}(a))$.
2. $x = a, a \in H$. We have two cases:
2.1 $a \in H_2 . \partial_H(a) = \delta = \partial_{H_1}(\delta) = \partial_{H_1}(\partial_{H_2}(a))$
2.2 $a \in H_1 - H_2 . \partial_H(a) = \delta = \partial_{H_1}(\partial_{H_2}(a))$
3. $x = \tau$. $\partial_H(\tau) = \tau = \partial_{H_1}(\tau) = \partial_{H_1}(\partial_{H_2}(\tau))$.
4. $x = yz$. $\partial_H(yz) = \partial_H(y)\partial_H(z) = \partial_{H_1}(\partial_{H_2}(y)).\partial_{H_1}(\partial_{H_2}(z)) =$
$= \partial_{H_1}(\partial_{H_2}(y)\partial_{H_2}(z)) = \partial_{H_1} \circ \partial_{H_2}(yz)$
5. $x = y + z$. $\partial_H(y+z) = \partial_H(y) + \partial_H(z) = \partial_{H_1}(\partial_{H_2}(y)) + \partial_{H_1}(\partial_{H_2}(z)) =$
$\partial_{H_1}(\partial_{H_2}(y) + \partial_{H_2}(z)) = \partial_{H_1} \circ \partial_{H_2}(y+z)$.

4.6 Proof of CA6 for all closed ACP$_\tau$-terms is entirely similar to 4.5.

4.7 Proof of CA7 for all closed ACP$_\tau$-terms is by induction on x. We have 5 cases:
1. $x = a, a \notin H, a \notin I$. $\tau_I \circ \partial_H(a) = \tau_I(a) = a = \partial_H(a) = \partial_H(\tau_I(a))$.
2. $x = a, a \in H, a \notin I$. $\tau_I \circ \partial_H(a) = \tau_I(\delta) = \delta = \partial_H(a) = \partial_H(\tau_I(a))$
3. $x = a, a \notin H, a \in I$. $\tau_I \circ \partial_H(a) = \tau_I(a) = \tau = \partial_H(\tau) = \partial_H(\tau_I(a))$.
4. $x = yz$. $\tau_I \circ \partial_H(yz) = \tau_I(\partial_H(y)\partial_H(z)) = \tau_I \circ \partial_H(y) . \tau_I \circ \partial_H(z) =$
$= \partial_H \circ \tau_I(y) . \partial_H \circ \tau_I(z) = \partial_H(\tau_I(y) \cdot \tau_I(z)) = \partial_H \circ \tau_I(yz)$.
5. $x = y + z$. $\tau_I \circ \partial_H(y+z) = \tau_I(\partial_H(y) + \partial_H(z)) = \tau_I \circ \partial_H(y) + \tau_I \circ \partial_H(z) =$
$= \partial_H \circ \tau_I(y) + \partial_H \circ \tau_I(z) = \partial_H(\tau_I(y) + \tau_I(z)) = \partial_H \circ \tau_I(y+z)$.

5. Example I. BAG || BAG = BAG

5.1 Suppose we have two bags, linked together as in fig. 1.

fig. 1.

let D be a finite set of data, let

$$\begin{cases} r_n(d) = \textit{receive } d \in D \text{ along channel } n \\ s_n(d) = \textit{send } d \in D \text{ along channel } n, \end{cases}$$

then we define a bag with input channel n and output channel m by:

$$B^{nm} = \sum_{d \in D} r_n(d)(s_m(d) \| B) \qquad \text{(see 3.7)}$$

5.2 Now we want to 'hide' the internal channel 2 above. We define:

1. *Communication:* $s_2(d) | r_2(d) = c_2(d)$ (for all $d \in D$) (d is *communicated* along channel 2), and all other communications are equal to δ;
2. *Encapsulation:* $H = \{s_2(d), r_2(d) \mid d \in D\}$ (all unsuccessful communications);
3. *Abstraction:* $I = \{c_2(d) \mid d \in D\}$ (all internal actions).

5.3 Theorem: $B^{13} = \tau_I \circ \partial_H (B^{12} \| B^{23})$.
(this is the meaning of the heading $BAG \| BAG = BAG$)

Proof. Let us first check that

$$\alpha(B^{13}) = \alpha(\tau_I \circ \partial_H (B^{12} \| B^{23})).$$

5.3.1 By 3.7 $\alpha(B^{13}) = \{r_1(d), s_3(d) \mid d \in D\}$.

5.3.2.1. $\alpha(\tau_I \circ \partial_H (B^{12} \| B^{23})) = \alpha(\partial_H (B^{12} \| B^{23})) - I \subseteq (\alpha(B^{12} \| B^{23}) - H) - I =$

$$= [\alpha(B^{12}) \cup \alpha(B^{23}) \cup \alpha(B^{12}) | \alpha(B^{23})] - (H \cup I) =$$

$$= [\{r_1(d), s_2(d) \mid d \in D\} \cup \{r_2(d), s_3(d) \mid d \in D\} \cup$$

$$\cup \{c_2(d) \mid d \in D\}] - (H \cup I) = \{r_1(d), s_3(d) \mid d \in D\}.$$

5.3.2.2. On the other hand

$$\alpha(\tau_I \circ \partial_H (B^{12} \| B^{23})) \supseteq \alpha(\pi_2(\tau_I \circ \partial_H (B^{12} \| B^{23}))) =$$

$$= \alpha(\pi_2(\tau_I(\sum_{d \in D} r_1(d) \cdot \partial_H(s_2(d) \| B^{12} \| B^{23})))) =$$

$$= \alpha(\sum_{d \in D} r_1(d) \cdot \pi_1 \circ \tau_I (\sum_{e \in D} r_1(e) \cdot \partial_H(s_2(d) \| s_2(e) \| B^{12} \| B^{23})$$

$$+ c_2(d).\partial_H(B^{12} \| s_3(d) \| B^{23}))) =$$

$$= \{r_1(d) \mid d \in D\} \cup \alpha(\sum_{d \in D} (\sum_{e \in D} r_1(e) +$$

$$+ \tau.\pi_1 \circ \tau_I (\sum_{f \in D} r_1(f).\partial_H(s_2(f) \| B^{12} \| s_3(d) \| B^{23})$$

$$+ s_3(d).\partial_H(B^{12} \| B^{23})))) =$$

$$= \{r_1(d) \mid d \in D\} \cup \alpha(\sum_{d \in D} (\sum_{f \in D} r_1(f) + s_3(d))) =$$

$$= \{r_1(d), s_3(d) \mid d \in D\}.$$

Combining 1 and 2 gives

$$\alpha(\tau_I \circ \partial_H (B^{12} \| B^{23})) = \{r_1(d), s_3(d) \mid d \in D\},$$

the desired result.
Now we give the proof of the theorem, and refer, for certain verifications, to the notes following the proof.

5.3.3 $\partial_H(B^{12}\|B^{23}) =$

$$= \sum_{d\in D} r_1(d)\cdot\partial_H(s_2(d)\|B^{12}\|B^{23}) =$$

(use standard concurrency, see 1.3)

$$\sum_{d\in D} r_1(d)\cdot\partial_H(B^{12}\|(s_2(d)\|B^{23})) =$$

$$= \text{(CA5)} \sum_{d\in D} r_1(d).\partial_H\circ\partial_{\{s_2(d)\}}(B^{12}\|(s_2(d)\|B^{23})) =$$

$$= \text{(CA1, 5.3.5)} \sum_{d\in D} r_1(d).\partial_H\circ\partial_{\{s_2(d)\}}(B^{12}\|\partial_{\{s_2(d)\}}(s_2(d)\|B^{23})) =$$

$$= \text{(CA5)} \sum_{d\in D} r_1(d).\partial_H(B^{12}\|\partial_{\{s_2(d)\}}(s_2(d)\|B^{23})) =$$

$$= \text{(5.3.10)} \sum_{d\in D} r_1(d)\cdot\partial_H(B^{12}\|(c_2(d)s_3(d))\|B^{23}) =$$

$$= \sum_{d\in D} r_1(d)\cdot\partial_H((c_2(d)s_3(d))\|(B^{12}\|B^{23})) =$$

$$= \text{(CA1, 5.3.6)} \sum_{d\in D} r_1(d).\partial_H((c_2(d)s_3(d))\|\partial_H(B^{12}\|B^{23})) =$$

$$= \text{(CA3, 5.3.7)} \sum_{d\in D} r_1(d)((c_2(d)s_3(d))\|\partial_H(B^{12}\|B^{23})).$$

5.3.4 Using this result, we get

$$\tau_I\circ\partial_H(B^{12}\|B^{23}) =$$

$$= \text{(5.3.3)} \sum_{d\in D} r_1(d).\tau_I((c_2(d)s_3(d))\|\partial_H(B^{12}\|B^{23})) =$$

= (apply CA2 twice, use 5.3.6)

$$\sum_{d\in D} r_1(d).\tau_I(\tau_I(c_2(d)s_3(d))\|\tau_I\circ\partial_H(B^{12}\|B^{23})) =$$

$$= \sum_{d\in D} r_1(d).\tau_I((\tau s_3(d))\|\tau_I\circ\partial_H(B^{12}\|B^{23})) =$$

$$= \text{(CA4, 5.3.8)} \sum_{d\in D} r_1(d)((\tau s_3(d))\|\tau_I\circ\partial_H(B^{12}\|B^{23})) =$$

$$= \text{(3.3.9)} \sum_{d\in D} r_1(d).\tau(s_3(d)\|\tau_I\circ\partial_H(B^{12}\|B^{23})) =$$

$$= \sum_{d\in D} r_1(d)(s_3(d)\|\tau_I\circ\partial_H(B^{12}\|B^{23})) =$$

$$= \text{(RSP) } B^{13}.$$

This finishes the proof of the theorem.

Now we prove some facts that were needed in the proof above. Let A be the set of atomic actions.

5.3.5 $\alpha(B^{12})\,|\,(\alpha(s_2(d)\|B^{23})\cap\{s_2(d)\})\subseteq$

$$\subseteq\{r_1(e),s_2(e)\,|\,e\in D\}\,|\,\{s_2(d)\} = \varnothing, \text{ for each } d\in D.$$

5.3.6 $\{c_2(d),s_3(d)\,|\,d\in D\}\,|\,A = \varnothing$ by 5.2.1.

5.3.7 $\alpha(c_2(d)s_3(d)\|\partial_H(B^{12}\|B^{23}))\cap H\subseteq$

$$\subseteq[\{c_2(d),s_3(d)\}\cup(A-H)\cup\{c_2(d),s_3(d)\}\,|\,(A-H)]\cap H =$$

$$= [A-H]\cap H = \varnothing, \text{ for each } d\in D.$$

5.3.8 $\alpha(\tau_I \circ \partial_H(B^{12} \| B^{23}) \| \tau s_3(d)) \cap I \subseteq$

$$\subseteq [(A-I) \cup \{s_3(d)\} \cup (A-I) \mid \{s_3(d)\}] \cap I =$$
$$= [A-I] \cap I = \varnothing, \text{ for each } d \in D$$

5.3.9 $(\tau x) \| y = \tau(x \| y)$. For all finite closed x and y, this is shown in Bergstra & Klop [6], 4.2.1.7. We assume this identity holds for infinite terms as well.

5.3.10 Let $d \in D$. Then
1. $(c_2(d) s_3(d)) \| B^{23} =$

$$= c_2(d)(s_3(d) \| B^{23}) + \sum_{e \in D} r_2(e)(s_3(e) \| (c_2(d) s_3(d)) \| B^{23}).$$

2. On the other hand

$$\partial_{\{s_2(d)\}}(s_2(d) \| B^{23}) =$$
$$= c_2(d).\partial_{\{s_2(d)\}}(s_3(d) \| B^{23}) + \sum_{e \in D} r_2(e) \cdot \partial_{\{s_2(d)\}}(s_3(e) \| s_2(d) \| B^{23}) =$$

= (by CA3, use the following 5.3.11)

$$c_2(d)(s_3(d) \| B^{23}) + \sum_{e \in D} r_2(e).\partial_{\{s_2(d)\}}(s_3(e) \| s_2(d) \| B^{23}) =$$

= (by CA1, argue like 5.3.5)

$$c_2(d)(s_3(d) \| B^{23}) + \sum_{e \in D} r_2(e) \cdot \partial_{\{s_2(d)\}}(s_3(e) \| \partial_{\{s_2(d)\}}(s_2(d) \| B^{23})) =$$

= (by CA3, use the following 5.3.12)

$$c_2(d)(s_3(d) \| B^{23}) + \sum_{e \in D} r_2(e)(s_3(e) \| \partial_{\{s_2(d)\}}(s_2(d) \| B^{23})).$$

Thus we see in 1. and 2. that

$$(c_2(d) s_3(d)) \| B^{23} \text{ and } \partial_{\{s_2(d)\}}(s_2(d) \| B^{23})$$

satisfy the same recursive equation. Therefore they are equal by RSP (see 2.10).

In 5.3.10, we needed the following two facts:
5.3.11 $\alpha(s_3(d) \| B^{23})) \cap \{s_2(d)\} =$

$$= [\{s_3(d)\} \cup \{r_2(e), s_3(e) \mid e \in D\} \cup \{s_3(d)\} \mid \{r_2(e), s_3(e) \mid e \in D\}]$$
$$\cap \{s_2(d)\} \subseteq [A - \{s_2(d)\}] \cap \{s_2(d)\} = \varnothing \text{ (for } d \in D).$$

5.3.12 $\alpha(s_3(e) \| \partial_{\{s_2(d)\}}(s_2(d) \| B^{23})) \cap \{s_2(d)\} \subseteq$

$$\subseteq (\{s_3(e)\} \cup (A - \{s_2(d)\}) \cup \{s_3(e)\} \mid A) \cap \{s_2(d)\} \subseteq$$
$$\subseteq (A - \{s_2(d)\}) \cap \{s_2(d)\} = \varnothing \text{ (for } d, e \in D).$$

6. Example II:

$$\boxed{BAG^{n} = BAG}$$

6.1 We generalize example I to the case where we have n bags linked together, as in fig. 2.

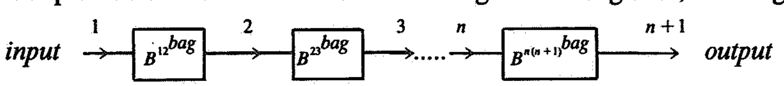

fig. 2.

To simplify notation, we will prove this in case $n=3$, but the proof works equally well if $n>3$. We use the notation of 5.1-2, so

6.2.1. Communication:
$s_n(d)\,|\,r_n(d)=c_n(d)$ $(n=2,3)$, all other communications are δ;

2. Encapsulation:
$H_n=\{s_n(d),\, r_n(d)\,|\,d\in D\}$ $(n=2,3)$, $H=H_2\cup H_3$;

3. Abstraction:
$I_n=\{c_n(d)\,|\,d\in D\}$ $(n=2,3)$, $I=I_2\cup I_3$.

6.3 Theorem. $B^{14} = \tau_I\circ\partial_H\,(B^{12}\|B^{23}\|B^{34})$.

Proof. First we need to check some alphabets. By 3.7 $\alpha(B^{12})=\{r_1(d),s_2(d)\,|\; d\in D\}$, so
6.3.1 $\alpha(B^{12})\,|\,(\alpha(B^{23}\|B^{34})\cap H_3)\subseteq\{r_1(d),s_2(d)\,|\,d\in D\}\,|\,H_3 = \varnothing$.
6.3.2 $\alpha(B^{12}\|\partial_{H_3}(B^{23}\|B^{34}))\cap H_3\subseteq[(A-H_3)\cup(A-H_3)\cup$

$$[(A-H_3)\,|\,(A-H_3)]\cap H_3 = [A-H_3]\cap H_3 = \varnothing.$$

6.3.3 $\alpha(B^{12})\,|\,(\alpha(\partial_{H_3}(B^{23}\|B^{34}))\cap I_3)\subseteq A\;|\,I_3 = \varnothing$. Now we can give the proof. Note that by 5.3 we have

$$B^{24} = \tau_{I_3}\circ\partial_{H_3}(B^{23}\|B^{34}) \text{ and}$$
$$B^{14} = \tau_{I_2}\circ\partial_{H_2}(B^{12}\|B^{24}).$$

Therefore $\tau_I\circ\partial_H(B^{12}\|B^{23}\|B^{34}) =$

$$= \text{(CA5, CA6) } \tau_{I_2}\circ\tau_{I_3}\circ\partial_{H_2}\circ\partial_{H_3}(B^{12}\|B^{23}\|B^{34}) =$$
$$= \text{(CA7) } \tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}\circ\partial_{H_3}(B^{12}\|B^{23}\|B^{34}) =$$
$$= \text{(CA1, 6.3.1) } \tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}\circ\partial_{H_3}(B^{12}\|\partial_{H_3}(B^{23}\|B^{34})) =$$
$$= \text{(CA3, 6.3.2) } \tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}(B^{12}\|\partial_{H_3}(B^{23}\|B^{34})) =$$
$$= \text{(CA2, 6.3.3) } \tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}(B^{12}\|\tau_{I_3}\circ\partial_{H_3}(B^{23}\|B^{34})) =$$
$$= \text{(5.3) } \tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}(B^{12}\|B^{24}) =$$
$$= \text{(CA4) } \tau_{I_2}\circ\partial_{H_2}(B^{12}\|B^{24}) =$$
$$= \text{(5.3) } B^{14}.$$

7 Example III:

$$\boxed{BAG.w.\,\varnothing\,\|BAG = \quad BAG}$$

7.1. In this example we consider a bag with test for empty. Such a bag can be defined by the following equations (notations as in 5.1):

$$\boxed{\begin{array}{ll} B\,\varnothing = & \sum_{d\in D}(r_1(d)B_d+s_2(\varnothing))B\,\varnothing \\ B_d = & s_2(d)+\sum_{e\in D}r_1(e)(B_e\,\|B_d) \end{array}}$$

To see that this indeed defines a bag with test for empty, consider the following

7.2 Lemma. $\partial_{\{s_2(\varnothing)\}}(B\,\varnothing)=B^{12}$ (notations from 5.1).

Proof. To prove this lemma, we need RSP (see 2.10), but here we need it for an *infinite* number of equations. In fact, we'll show that for all multisets G, consisting of elements of D, we have

$$\boxed{B^{12} \| (\underset{e \in G}{\|} s_2(e)) = (\underset{e \in G}{\|} B_e).\partial_{\{s_2(\varnothing)\}}(B\,\varnothing) \qquad (*)}$$

(for $G = \varnothing$, we define $B^{12} \| (\underset{e \in \varnothing}{\|} s_2(e)) = B^{12}$ and

$$(\underset{e \in \varnothing}{\|} B_e) \cdot \partial_{\{s_2(\varnothing)\}}(B\,\varnothing) = \partial_{\{s_2(\varnothing)\}}(B\,\varnothing),$$

so the lemma follows immediately from (*).)
To show (*), we consider two cases:
case 1: $G = \varnothing$. Then

$$\partial_{\{s_2(\varnothing)\}}(B\,\varnothing) = \sum_{d \in D} (r_1(d) \cdot \partial_{\{s_2(\varnothing)\}}(B_d) + \delta).\partial_{\{s_2(\varnothing)\}}(B\,\varnothing) =$$

$$= \sum_{d \in D} r_1(d) B_d \partial_{\{s_2(\varnothing)\}}(B\,\varnothing) =$$

(by CA3, see verification below)

$$= \text{(RSP)} \sum_{d \in D} r_1(d)(B^{12} \| s_2(d)) =$$

$$= (5.1)\ B^{12}.$$

The verification of the third step above:

$$\alpha(B_d) = \{s_2(d)\} \cup \{r_1(e) \mid e \in D\} \cup \bigcup_{e \in D} \alpha(B_e \| B_d) =$$

$$= \{s_2(d)\} \cup \{r_1(e) \mid e \in D\} \cup \bigcup_{e \in D} \alpha(B_e) \cup \alpha(B_d) \cup$$

$$\cup \bigcup_{e \in D} \alpha(B_e) \mid \alpha(B_d).$$

Since $\beta(B_d)$ is the least fixed point of this equation, it follows easily that

$$\beta(B_d) = \{s_2(e), r_1(e) \mid e \in D\}.$$

Thus $\beta(B_d) \cap \{s_2(\varnothing)\} = \varnothing$, whence $\alpha(B_d) \cap \{s_2(\varnothing)\} = \varnothing$.

case 2: $G \neq \varnothing$. Then we need the Expansion Theorem (see Bergstra & Tucker [8]):

let $\vec{X}^i = \underset{n \in \{1,\ldots,k\} - \{i\}}{\|} X_i$, $\vec{X}^{i,j} = \underset{n \in \{1,\ldots,k\} - \{i,j\}}{\|} X_i$, then

$$\boxed{\text{(ET)} \quad X_1 \| \cdots \| X_k = \sum_{1 \leqslant i \leqslant k} X_i \,⌊\!\!⌊\, \vec{X}^i + \sum_{1 \leqslant i < j \leqslant k} (X_i \mid X_j) \,⌊\!\!⌊\, \vec{X}^{i,j}}$$

The proof of ET uses the Handshaking Axiom (see 3.6). The Expansion Theorem says, that if we have a merge of a number of terms, then we can start with an action in one of the terms or with a communication between two of the terms. Using that here, we get

$$B^{12} \| (\underset{e \in G}{\|} s_2(e)) = \sum_{d \in G} s_2(d)(B^{12} \| (\underset{e \in G - \{d\}}{\|} s_2(e))) +$$

$$+ \sum_{d \in D} r_1(d)(B^{12} \| (\underset{e \in G \cup \{d\}}{\|} s_2(e))) =$$

$$= \text{(RSP)} \sum_{d \in G} s_2(d)(\underset{e \in G - \{d\}}{\|} B_e) \cdot \partial_{\{s_2(\varnothing)\}}(B\,\varnothing) +$$

$$+ \sum_{d \in D} r_1(d)(\underset{e \in G \cup \{d\}}{\|} B_e).\partial_{\{s_2(\varnothing)\}}(B\,\varnothing) =$$

$$= (\mathrm{ET})\,(\underset{e\in G}{\|} B_e)\cdot\partial_{\{s_2(\varnothing)\}}(B\varnothing).$$

7.3 Now suppose we want to link this bag with empty test to a regular bag. Between the two, we interpose a one-place buffer, that "forgets" the empty test, i.e. a

$$\boxed{T\varnothing \;=\; \sum_{d\in D}(r_2(d)s_3(d)+r_2(\varnothing))T\varnothing}$$

Thus we have the situation of fig. 3

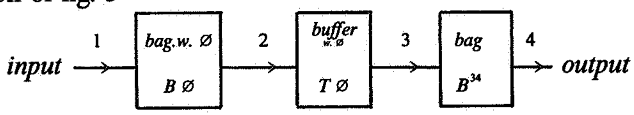

fig. 3

7.4 We define

1. *Communication:* as in 6.2, also $s_2(\varnothing)\,|\,r_2(\varnothing)=c_2(\varnothing)$
2. *Encapsulation:* as in 6.2, also $H_\varnothing=\{s_2(\varnothing),r_2(\varnothing)\}$, and $H=H_2\cup H_3\cup H_\varnothing$.

3. *Abstraction:* as in 6.2, but $I=I_2\cup I_3\cup\{c_2(\varnothing)\}$.

7.5 Theorem

$$\boxed{\tau_I\circ\partial_H(B\varnothing\|T\varnothing\|B^{34}) = \tau B^{14}.}$$

(this is the statement refered to in the heading of this paragraph.) Before we can prove this theorem, we will need a number of lemmas. First, define a (regular) one-place buffer T by:

$$\boxed{T = \sum_{d\in D} r_2(d)s_3(d)T}$$

7.6 Lemma:

$$\partial_{\{r_2(\varnothing)\}}(T\varnothing) = T$$

Proof:

$$\partial_{\{r_2(\varnothing)\}}(T\varnothing) = \sum_{d\in D}(r_2(d)s_3(d)+\delta).\partial_{\{r_2(\varnothing)\}}(T\varnothing) =$$

$$= \sum_{d\in D} r_2(d)s_3(d).\partial_{\{r_2(\varnothing)\}}(T\varnothing).$$

Now use RSP.

7.7 Define

$$P = \partial_{H_2\cup H_\varnothing}(B\varnothing\|T\varnothing)$$

Lemma: There is a Q such that $P = c_2(\varnothing)\cdot P+QP$ and $c_2(\varnothing)\notin\alpha(Q)$.

Proof: We will give an infinite recursive specification of Q. This specification has variables $P(G)$ and $P(G,d)$, for each multiset G consisting of elements of D, and each $d\in D$.
(The intuitive meaning is that $P(G)$ describes the bag with contents G, the buffer empty; and $P(G,d)$ describes the bag with contents G, and the buffer filled with d.) We have the following equations:
case 1: $G=\varnothing$.

$$Q = \sum_{d\in D} r_1(d)P(\{d\}),\quad P(\varnothing) = P$$

$$P(\varnothing,d) = s_3(d) + \sum_{e\in D} r_1(e)P(\{e\},d)$$

case 2: $G \neq \varnothing$. Then

$$P(G) = \sum_{d \in G} c_2(d)P(G-\{d\},d) + \sum_{d \in D} r_1(d)P(G \cup \{d\}), \text{and}$$
$$P(G,d) = s_3(d)P(G) + \sum_{e \in D} r_1(e)P(G \cup \{e\},d).$$

Then we have

$$P = \partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing) =$$
$$= c_2(\varnothing)\partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing) + \sum_{d \in D} r_1(d)\partial_{H_2 \cup H_\varnothing}(B_d B\varnothing \| T\varnothing)$$
$$= c_2(\varnothing).P + \sum_{d \in D} r_1(d)P(\{d\})P = c_2(\varnothing)P + QP,$$

which follows from the following two claims:

$$a: \quad \partial_{H_2 \cup H_\varnothing}((\|_{e \in G} B_e)B\varnothing \| T\varnothing) = P(G).P$$
$$b: \quad \partial_{H_2 \cup H_\varnothing}((\|_{e \in G} B_e)B\varnothing \| s_3(d)T\varnothing) = P(G,d).P$$

These we again prove by the infinite RSP.
case 1. $G = \varnothing . a$: is shown above

$$b: \quad \partial_{H_2 \cup H_\varnothing}(B\varnothing \| s_3(d)T\varnothing) =$$
$$= s_3(d)\partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing) + \sum_{e \in D} r_1(e)\cdot\partial_{H_2 \cup H_\varnothing}(B_e B\varnothing \| s_3(d)T\varnothing) =$$
$$= (\text{RSP})\ s_3(d).P + \sum_{e \in D} r_1(e).P(\{e\},d).P =$$
$$= (s_3(d) + \sum_{e \in D} r_1(e)\cdot P(\{e\},d))P = P(\varnothing,d).P$$

case 2: $G \neq \varnothing . a$: $\partial_{H_2 \cup H_\varnothing}((\|_{e \in G} B_e)B\varnothing \| T\varnothing) =$

$$= (\text{ET}) \sum_{d \in G} c_2(d)\cdot\partial_{H_2 \cup H_\varnothing}((\|_{e \in G-\{d\}} B_e)B\varnothing \| s_3(d)T\varnothing) +$$
$$+ \sum_{d \in D} r_1(d)\cdot\partial_{H_2 \cup H_\varnothing}((\|_{e \in G \cup \{d\}} B_e)B\varnothing \| T\varnothing) =$$
$$= (\text{RSP}) \sum_{d \in G} c_2(d)P(G-\{d\},d).P + \sum_{d \in D} r_1(d).P(G \cup \{d\}).P$$
$$= P(G).P, \text{ and}$$

$$b: \quad \partial_{H_2 \cup H_\varnothing}((\|_{e \in G} B_e)B\varnothing \| s_3(d)T\varnothing) =$$
$$= (\text{ET})\ s_3(d).\partial_{H_2 \cup H_\varnothing}((\|_{e \in G} B_e)B\varnothing \| T\varnothing) +$$
$$+ \sum_{f \in D} r_1(f)\cdot\partial_{H_2 \cup H_\varnothing}((\|_{e \in G \cup \{f\}} B_e)B\varnothing \| s_3(d)T\varnothing) =$$
$$= (\text{RSP})\ s_3(d).P(G).P + \sum_{f \in D} r_1(f)\cdot P(G \cup \{f\},d).P$$
$$= P(G,d).P.$$

Next, it is not much trouble to show that for all multisets G and all $d \in D$,

$$\beta(P(G)) = \{r_1(d), c_2(d), s_3(d) \mid d \in D\} \quad (G \neq \varnothing), \text{ and}$$
$$\beta(P(G,d)) = \{r_1(d), c_2(d), s_3(d) \mid d \in D\}.$$

Thus, $\beta(Q) = \{r_1(d), s_2(d), s_3(d) \mid d \in D\}$, whence $c_2(\varnothing) \notin \beta(Q)$ and $c_2(\varnothing) \notin \alpha(Q)$.

7.8 Note: if we can use *priorities* on atomic actions (see Baeten, Bergstra & Klop [1]), then Q can be defined by a finite specification, as follows. If all actions $s_3(d)$ (for $d \in D$) have priority over all $c_2(e)$ (for $e \in D$), and θ implements this priority (i.e. $\theta(s_3(d)x + c_2(e)y) = s_3(d)\theta(x)$), we can define

$$C_d = c_2(d)T_d + \sum_{e \in D} r_1(e)(C_e \| C_d) \quad (\text{for } d \in D)$$

$$T_d = s_3(d) + \sum_{e \in D} r_1(e)(T_d \| C_e) \quad (\text{for } d \in D)$$

$$Q = \sum_{d \in D} r_1(d) \cdot \theta(C_d).$$

7.9 Lemma. $\tau_{\{c_2(\varnothing)\}}(P) = \tau.\partial_{\{c_2(\varnothing)\}}(P)$.

Proof. By 7.7, we have $P = c_2(\varnothing)P + QP$. Applying KFAR (see 2.17) to this equation, we get

$$\tau_{\{c_2(\varnothing)\}}(P) = \tau.\tau_{\{c_2(\varnothing)\}}(QP) =$$
$$= \tau.\tau_{\{c_2(\varnothing)\}}(Q) \cdot \tau_{\{c_2(\varnothing)\}}(P) =$$
$$= \tau.Q.\tau_{\{c_2(\varnothing)\}}(P)$$

by CA4, since $c_2(\varnothing) \notin \alpha(Q)$. On the other hand

$$\tau.\partial_{\{c_2(\varnothing)\}}(P) = \tau.\partial_{\{c_2(\varnothing)\}}(c_2(\varnothing)P + QP) =$$
$$= \tau(\delta + \partial_{\{c_2(\varnothing)\}}(QP)) =$$
$$= \tau.\partial_{\{c_2(\varnothing)\}}(Q) \cdot \partial_{\{c_2(\varnothing)\}}(P) =$$
$$= (\text{CA3})\ \tau.Q.\tau\partial_{\{c_2(\varnothing)\}}(P),$$

so by RSP $\tau_{\{c_2(\varnothing)\}}(P) = \tau.\partial_{\{c_2(\varnothing)\}}(P)$.

7.10 Lemma.

$$\tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing) = \tau.\tau_{I_2} \circ \partial_{H_2}(B^{12} \| T)$$

Proof.

$$\tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing) =$$
$$= (\text{CA6})\ \tau_{I_2} \circ \tau_{\{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing) =$$
$$= (7.9)\ \tau_{I_2}(\tau.\partial_{\{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_\varnothing}(B\varnothing \| T\varnothing)) =$$
$$= (\text{CA5})\ \tau.\tau_{I_2} \circ \partial_{H_2} \circ \partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(B\varnothing \| T\varnothing) =$$

= (CA1, see note 1 below)

$$\tau.\tau_{I_2} \circ \partial_{H_2} \circ \partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(\partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(B\varnothing) \| \partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(T\varnothing)) =$$
$$= (\text{CA5})\ \tau.\tau_{I_2} \circ \partial_{H_2} \circ \partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(\partial_{\{c_2(\varnothing), r_2(\varnothing)\}} \circ \partial_{\{s_2(\varnothing)\}}(B^{12}) \| \partial_{\{c_2(\varnothing), s_2(\varnothing)\}} \circ \partial_{\{r_2(\varnothing)\}}(T\varnothing)) =$$
$$= (7.2, 7.6)\ \tau.\tau_{I_2}\partial_{H_2} \circ \partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(\partial_{\{c_2(\varnothing), r_2(\varnothing)\}}(B^{12}) \| \partial_{\{c_2(\varnothing), s_2(\varnothing)\}}(T)) =$$

= (CA3, see note 2 below)

$$\tau.\tau_{I_2} \circ \partial_{H_2} \circ \partial_{\{c_2(\varnothing)\} \cup H_\varnothing}(B^{12} \| T) =$$

= (CA3, see note 3 below)

$$\tau.\tau_{I_2} \circ \partial_{H_2}(B^{12} \| T).$$

Notes:

1. This is because $\{c_2(\varnothing)\} \cup H_{\varnothing}$ is closed under communication.
2. $\alpha(B^{12}) = \{r_1(d), s_2(d) \mid d \in D\}$ by 1.25, and $\alpha(T) = \{r_2(d), s_3(d) \mid d \in D\}$ is easily proved.
3. $\alpha(B^{12} \| T) = \alpha(B^{12}) \cup \alpha(T) \cup \alpha(B^{12}) \mid \alpha(T) = \{r_1(d),\ s_2(d),\ c_2(d),\ r_2(d),\ s_3(d) \mid d \in D\}$.

7.11 Lemma $\tau_{I_3} \circ \partial_{H_3}(T \| B^{34}) = B^{24}$

Proof.

$$\tau_{I_3} \circ \partial_{H_3}(T \| B^{34}) =$$

$$= \tau_{I_3}\Big(\sum_{d \in D} r_2(d) \cdot \partial_{H_3}((s_3(d)T) \| B^{34})\Big) =$$

$$= \sum_{d \in D} r_2(d) \cdot \tau_{I_3}(c_3(d) \cdot \partial_{H_3}(T \| B^{34} \| s_4(d))) =$$

$=$ (CA1, note that $A \mid \{s_4(d)\} = \varnothing$)

$$\sum_{d \in D} r_2(d) . \tau \tau_{I_3} \circ \partial_{H_3}(\partial_{H_3}(T \| B^{34}) \| s_4(d)) =$$

$$= \text{(CA3)} \sum_{d \in D} r_2(d) \cdot \tau_{I_3}(\partial_{H_3}(T \| B^{34}) \| s_4(d)) =$$

$$= \text{(CA2)} \sum_{d \in D} r_2(d) \tau_{I_3}(\tau_{I_3} \circ \partial_{H_3}(T \| B^{34}) \| s_4(d)) =$$

$$= \text{(CA4)} \sum_{d \in D} r_2(d)(\tau_{I_3} \circ \partial_{H_3}(T \| B^{34}) \| s_4(d)) =$$

$$= \text{(RSP) } B^{24}.$$

7.12 Now we can finally give the proof of theorem 7.5:

$$\tau_I \circ \partial_H(B\varnothing \| T\varnothing \| B^{34}) = \tau B^{14}.$$

Proof: $\tau_I \circ \partial_H(B\varnothing \| T\varnothing \| B^{34}) =$

$$= \text{(CA5, CA6) } \tau_{I_3} \circ \tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_3} \circ \partial_{H_2 \cup H_{\varnothing}}(B\varnothing \| T\varnothing \| B^{34}) =$$

$$= \text{(CA7) } \tau_{I_3} \circ \partial_{H_3} \circ \tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_{\varnothing}}((B\varnothing \| T\varnothing) \| B^{34}) =$$

$=$ (CA1, see note 1 below)

$$\tau_{I_3} \circ \partial_{H_3} \circ \tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_{\varnothing}}(\partial_{H_2 \cup H_{\varnothing}}(B\varnothing \| T\varnothing) \| B^{34}) =$$

$=$ (CA3, see note 1 below)

$$\tau_{I_3} \circ \partial_{H_3} \circ \tau_{I_2 \cup \{c_2(\varnothing)\}}(\partial_{H_2 \cup H_{\varnothing}}(B\varnothing \| T\varnothing) \| B^{34}) =$$

$=$ (CA2, see note 1 below)

$$\tau_{I_3} \circ \partial_{H_3} \circ \tau_{I_2 \cup \{c_2(\varnothing)\}}(\tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_{\varnothing}}(B\varnothing \| T\varnothing) \| B^{34}) =$$

$=$ (CA4, see note 1 below)

$$\tau_{I_3} \circ \partial_{H_3}(\tau_{I_2 \cup \{c_2(\varnothing)\}} \circ \partial_{H_2 \cup H_{\varnothing}}(B\varnothing \| T\varnothing) \| B^{34}) =$$

$$= \text{(7.10) } \tau_{I_3} \circ \partial_{H_3}(\tau . \tau_{I_2} \circ \partial_{H_2}(B^{12} \| T) \| B^{34}) =$$

$$= \text{(5.3.9) } \tau . \tau_{I_3} \circ \partial_{H_3}(\tau_{I_2} \circ \partial_{H_2}(B^{12} \| T) \| B^{34}) =$$

$=$ (CA3, CA4, see note 2 below)

$$\tau . \tau_{I_3} \circ \partial_{H_3} \circ \tau_{I_2} \circ \partial_{H_2}(\tau_{I_2} \circ \partial_{H_2}(B^{12} \| T) \| B^{34}) =$$

= (CA1, CA2, see note 3 below)

$$\tau.\tau_{I_3}\circ\partial_{H_3}\circ\tau_{I_2}\circ\partial_{H_2}(B^{12}\|T\|B^{34}) =$$

$$= \text{(CA7) } \tau.\tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}\circ\partial_{H_3}(B^{12}\|T\|B^{34}) =$$

= (CA1, CA2, see note 4 below)

$$\tau.\tau_{I_2}\circ\partial_{H_2}\circ\tau_{I_3}\circ\partial_{H_3}(B^{12}\|\tau_{I_3}\circ\partial_{H_3}(T\|B^{34})) =$$

= (CA3, CA4, see note 4 below)

$$\tau.\tau_{I_2}\circ\partial_{H_2}(B^{12}\|\tau_{I_3}\circ\partial_{H_3}(T\|B^{34})) =$$

$$= (7.11)\ \tau.\tau_{I_2}\circ\partial_{H_2}(B^{12}\|B^{24}) =$$

$$= (5.3)\ \tau.B^{14}.$$

Notes: 1. By 3.7, $\alpha(B^{34}) = \{r_3(d), s_4(d) \mid d \in D\}$. Thus $\alpha(B^{34}) \mid (A - \{s_3(d) \mid d \in D\}) = \varnothing$, and also

$$((A - (H_2 \cup H_\varnothing)) \cup \alpha(B^{34})) \cap (H_2 \cup H_\varnothing) = \varnothing,$$

$$((A - (I_2 \cup \{c_2(\varnothing)\})) \cup \alpha(B^{34})) \cap (I_2 \cup \{c_2(\varnothing)\}) = \varnothing$$

2. Likewise $((A - H_2) \cup \alpha(B^{34})) \cap H_2 = \varnothing$ and

$$((A - I_2) \cup \alpha(B^{34})) \cap I_2 = \varnothing$$

3. $\alpha(B^{12}\|T) \mid \alpha(B^{34}) =$

$$= (\{r_1(d), s_2(d) \mid d \in D\} \cup \{r_2(d), s_3(d) \mid d \in D\} \cup \{c_2(d) \mid d \in D\}) \mid$$

$$\mid \{r_3(d), s_4(d) \mid d \in D\} = \{c_3(d) \mid d \in D\}.$$

The result is $\varnothing$ of we intersect $\alpha(B^{12}\|T)$ with I_2 or H_2.

4. $\alpha(B^{12}) \mid \alpha(T\|B^{34}) =$

$$= \{r_1(d), s_2(d) \mid d \in D\} \mid \{r_2(d), s_3(d), c_3(d), r_3(d), s_4(d) \mid d \in D\} =$$

$$= \{c_2(d) \mid d \in D\}.$$

The result is $\varnothing$ if we intersect $\alpha(T\|B^{34})$ with I_3 or H_3.

References

[1] BAETEN, J.C.M., J.A. BERGSTRA & J.W. KLOP, *Syntax and defining equations for an interrupt mechanism in process algebra,* Report CS-R85.., Centrum voor Wiskunde en Informatica, Amsterdam 1985.

[2] DE BAKKER, J.W. & J.I. ZUCKER, *Processes and the denotational semantics of concurrency,* Information and Control, 54 (1/2), 1982, pp. 70-120.

[3] BERGSTRA, J.A. & J.W. KLOP, *The algebra of recursively defined processes and the algebra of regular processes,* Report IW 235/83, Centrum voor Wiskunde en Informatica, Amsterdam 1983.

[4] BERGSTRA, J.A. & J.W. KLOP, *Algebra of communicating processes with abstraction,* Report CS-R8403, Centrum voor Wiskunde en Informatica, Amsterdam 1984, to appear in TCS.

[5] BERGSTRA, J.A. & J.W. KLOP, *Verification of an alternating bit protocol by means of process algebra,* Report CS-R8404, Centrum voor Wiskunde en Informatica, Amsterdam 1984.

[6] BERGSTRA, J.A. & J.W. KLOP, *Process algebra for synchronous communication,* Information and Control, 60 (1/3), 1984, pp. 109-137.

[7] BERGSTRA, J.A. & J.W. KLOP, *Algebra of communicating processes,* Proceedings of the CWI symposium Mathematics and Computer Science, eds. J.W. de Bakker, M. Hazewinkel & J.K. Lenstra, to appear.

[8] BERGSTRA, J.A. & J.V. TUCKER, *Top-down design and the algebra of communicating processes,* Report CS-R8401, Centrum voor Wiskunde en Informatica, Amsterdam 1984, to appear in SCP.

[9] BROOKES, S., C. HOARE & W. ROSCOE, *A theory of communicating sequential processes,* JACM, 31 (3), 1984, pp. 560-599.

[10] HENNESSY, M. & G. PLOTKIN, *A term model for CCS,* Proc. 9th MFCS, Poland (1980), Springer LNCS 88.

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

[12] OLDEROG, E.-R., *Specification oriented programming,* to appear.

[13] SIFAKIS, J., *Property preserving homomorphisms of transition systems,* Proc. Logics of Programs (1983), Springer LNCS 164, 1984.