Process Algebra with Combinators

Jan A. Bergstra[1,2] Inge Bethke[1] Alban Ponse[1]

[1] *University of Amsterdam, Programming Research Group*
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

[2] *Utrecht University, Department of Philosophy*
Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

E-mail: `janb@fwi.uva.nl` - `inge@fwi.uva.nl` - `alban@fwi.uva.nl`

Abstract

We introduce typed combinatory process algebra, a system combining process algebra with types and combinators. We describe its syntax and semantics, and by way of example, verify within this framework the Simple Alternating Bit Protocol.

Key Words & Phrases: protocol verification, process algebra, typed combinatory logic.
1991 Mathematics Subject Classification: 69C20, 69M10, 03B15, 03B40.

Note: The first author acknowledges the support of ESPRIT Basic Research Action CONFER no. 6454.

Contents

1 Introduction

System specification and verification in process algebra always combines data structuring (e.g. using abstract data types) and control structuring which is done by means of the primitives of a suitable process algebra. There are several languages that link a notation suggested by process algebra to some abstract data type notation. We mention: LOTOS ([BB87],[B88]), PSF ([MV90],[M91]), μCRL ([GP90], [GP91]). In each of these cases a rudimentary form of typed λ-calculus is used to organize the distribution of data within a process expression. In this paper we intend to clarify in detail the type structure of data dependent actions and processes. In addition we propose to employ typed combinators (cf. e.g. [S24],[HS86]) in order to stay entirely within typed equational logic.

As an illustration SABP, the Simple Alternating Bit Protocol, taken from [P85] and adopted to ACP syntax in [BKO87] is verified in a purely equational way. This improves all previous verifications, e.g. the ones in [B90], by not using conditional equations. Our analysis of SABP essentially uses the single exit iteration operator (Kleene star) of [BBP93] and the corresponding version of Koomen's fair abstraction rule.

As to the relevance of this work, we state the following

(i) we have developed a purely equational style of reasoning about protocols in process algebra that applies at least in some simple cases;

(ii) we have elaborated on a reasonable typing schema that can underly process specifications such as can be given e.g. in μCRL;

(iii) we think that much work on process algebra comprises rudimentary forms of type theory and typed λ-calculus that could be made explicit and phrased in terms of existing type theoretical primitives. In doing so, process algebra is reduced to its essential content which will improve its clarity. Here is an entire research programme visible of which the present paper is one of the more obvious steps.

The paper is organized as follows. In Section 2 we collect the basic definitions concerning typed combinatory logic, and its accompanying Extensionality Theorem. Our exposition is based in part on [HS86], [CF58] and [S67]. In Section 3 we introduce the formal language of combinatory process algebra including the operator Σ for arbitrary sum formation, and give an axiomatization based on $\mathrm{ACP}_\tau{}^*$. Moreover, we prove a number of logical consequences of this axiomatization which we then apply in the next section. Section 4 is devoted entirely to the verification of SABP. Here we enforce due rigour in order to illustrate in detail the ins and outs of the system of combinatory process algebra. In Section 5 we describe informally a natural semantics of combinatory process algebra, and finally, in Section 6, we discuss briefly how the state operator can be fitted into this framework.

2 Types and combinators

The type structure that seems to be appropriate for various process algebra systems is the polymorphic type structure generated by a partially ordered set of basic types $(\mathcal{B}, \subseteq)$ that contains the sets of core atoms, atoms, processes and data, and their natural subset relation.

Definition 2.1. $(\mathcal{T}(\mathcal{B}), \leqslant)$
Let $\mathcal{B} = (B, \subseteq)$ be a partially ordered set containing $(\{A^c, A, P, D\}, A^c \subseteq A \subseteq P)$.

(i) The set of types, $\mathcal{T}(\mathcal{B})$, is defined as follows.

$$\mathcal{T}(\mathcal{B}) := B \mid \mathcal{T}(\mathcal{B}) \to \mathcal{T}(\mathcal{B})$$

(ii) The pre-order on $\mathcal{T}(\mathcal{B})$, $\preccurlyeq$, is defined by the following clauses.

(a) If $\beta \subseteq \beta'$, then $\beta \preccurlyeq \beta'$, for all $\beta, \beta' \in B$.

(b) If $\sigma \preccurlyeq \sigma'$ and $\rho \preccurlyeq \rho'$, then $\sigma' \to \rho \preccurlyeq \sigma \to \rho'$.

(iii) The subtype relation on $\mathcal{T}(\mathcal{B})$, $\leqslant$, is the reflexive, transitive closure of $\preccurlyeq$.

A, the set of atoms, is intended to comprise the core atoms together with deadlock and the silent step. Each compound type $\sigma \to \rho$ is intended to denote the set of functions from σ to ρ. More precisely, these are functions whose domain is the set denoted by σ, and whose range is a subset of the set denoted by ρ. For example, $D \to P$ denotes the set of functions from data to processes, and, written linearly, $D \to (A^c \to P)$ denotes the set of binary functions mapping pairs consisting of a datum and a core atom to processes.

The order on compound types is, in a natural way, induced by the order on basic types. To give an

example, observe that, with the interpretation of types given so far, every core atom is an atom and every atom is a process. A function from processes to core atoms, for example, corresponds therefore uniquely to a function from atoms to atoms and to a function from atoms to processes: namely to its restriction to A. This defines an embedding $\upharpoonright A$ from $P \to A^c$ to $A \to A$ and $A \to P$. In this way, we may consider $P \to A^c$ to be a subtype of $A \to A$ and $A \to P$.
If $\sigma \preccurlyeq A$ ($\sigma \preccurlyeq P$), we say that $\rho_1 \to (\rho_2 \to (\cdots(\rho_n \to \sigma)\cdots)$ is an *atom valued type* (a *process valued type*).

A $\mathcal{T}(\mathcal{B})$-typed signature $\Sigma = (\mathcal{B}, \mathcal{F})$ consists of a poset $\mathcal{B}$ of basic types and a set $\mathcal{F}$ of function symbols, each with a fixed type in $\mathcal{T}(\mathcal{B})$. The set of terms of typed combinatory logic over Σ, $T(\Sigma)$, is the set of expressions generated in the following way: For each type $\rho \in \mathcal{T}(\mathcal{B})$, we assume that there is a denumerable infinity of variables $V_\rho = \{x^\rho, y^\rho, z^\rho, \cdots\}$ of that type; we also assume, for all types $\rho, \rho', \rho'' \in \mathcal{T}(\mathcal{B})$, that there are combinators I_ρ of type $\rho \to \rho$, $K_{\rho,\rho'}$ of type $\rho \to (\rho' \to \rho)$, and $S_{\rho,\rho',\rho''}$ of type $(\rho \to (\rho' \to \rho'')) \to ((\rho \to \rho') \to (\rho \to \rho''))$. From these basic expressions together with the function symbols in $\mathcal{F}$ we build up terms inductively.

Definition 2.2. ($T(\Sigma)$)
Let $\Sigma = (\mathcal{B}, \mathcal{F})$ be a $\mathcal{T}(\mathcal{B})$-typed signature. Then we define $T(\Sigma) = \bigcup_{\rho \in \mathcal{T}(\mathcal{B})} T(\Sigma)_\rho$ as follows. For all $\rho, \rho', \rho'', \sigma \in \mathcal{T}(\mathcal{B})$,

(i) $V_\rho \subseteq T(\Sigma)_\rho$

(ii) $I_\rho \in T(\Sigma)_{\rho \to \rho}$, $K_{\rho,\rho'} \in T(\Sigma)_{\rho \to (\rho' \to \rho)}$, $S_{\rho,\rho',\rho''} \in T(\Sigma)_{(\rho \to (\rho' \to \rho'')) \to ((\rho \to \rho') \to (\rho \to \rho''))}$

(iii) If $f \in \mathcal{F}$ is of type ρ, then $f \in T(\Sigma)_\rho$.

(iv) If $t \in T(\Sigma)_{\rho \to \sigma}$ and $t' \in T(\Sigma)_{\rho'}$ with $\rho' \leqslant \rho$, then $tt' \in T(\Sigma)_\sigma$.

We write $t : \rho$ or t^ρ to indicate that t is a term of type ρ. If $t : \rho$, $t' : \rho'$, we say that t, t' are *compatible* iff there exists σ with $\rho \leqslant \sigma$, $\rho' \leqslant \sigma$, and *incompatible* iff no such σ exists. Type super- and subscripts will be often omitted when clear from the context. Parentheses will be omitted too in such a way that, for example, $tt't''$ denotes $(tt')t''$. This convention is called *association to the left.*

Combinatory logic is an equational theory. Its fundamental axioms, that define the applicative behaviour of the *elementary identificator*[1] I, the *elementary cancellator* K, and the anonymous combinator S, are listed in Table 1. In addition, there are the usual axioms and rules of equational logic with substitution respecting subtypes. That is, a substitution is now a function $\alpha : \bigcup_{\sigma \in \mathcal{T}(\mathcal{B})} V_\sigma \to T(\Sigma)$ such that $\alpha[V_\sigma] \subseteq \bigcup_{\sigma' \leqslant \sigma} T(\Sigma)_{\sigma'}$.

(I_σ)	$I_\sigma x$	$=$	x
($\mathrm{K}_{\sigma,\sigma'}$)	$K_{\sigma,\sigma'} xy$	$=$	x
($\mathrm{S}_{\sigma,\sigma',\sigma''}$)	$S_{\sigma,\sigma',\sigma''} xyz$	$=$	$xz(yz)$

Table 1: The axioms of combinatory logic.

With the aid of the fundamental combinators and their defining equations, one can define *abstraction terms* $[x]\,t$ for each x and t, with the property that

$$([x]\,t)t' = t[x := t']$$

where $t[x := t']$ denotes the result of substituting t' for every occurrence of x in t. In contrast to λ in the λ-calculus, $[x]$ is not part of the formal language of Σ-terms; $[x]\,t$ will be a combination of I's, K's and S's and parts of t, built up as follows.

[1]The nomenclature is taken from [CF58]

Definition 2.3. (Abstraction)
Let $t:\sigma$ and $x \in V_\rho$. The $[x]\,t:\rho\rightarrow\sigma$ is defined by the following clauses.

(i) If $t \equiv x$, then $[x]\,t \triangleq I$.

(ii) If t is a variable distinct from x, or $t \in \mathcal{F}\cup\{I,K,S\}$, then $[x]\,t \triangleq Kt$.

(iii) If $t \equiv t't''$, then $[x]\,t \triangleq S([x]\,t')([x]\,t'')$.

The preceding definition can be generalized for several variables:

$$[x_0,\ldots,x_{n+1}]\,t \triangleq [x_0,\ldots,x_n]\,[x_{n+1}]\,t.$$

Note that abstraction variables do not occur anymore in the abstraction terms.

Combinatory logic is not extensional. That is, terms having the same applicative behaviour are not in general proved equal. In order to obtain an extensional theory, the following axioms, taken from [S67], have to be added. Note that the terms in axioms (E1)-(E5) do not contain variables.

(E1)	$[x,y]\,S(S(KK)x)y$	$=$	K
(E2)	$[x,y,z]\,S(S(S(KS)x)y)z$	$=$	$[x,y,z]\,S(Sxz)(Syz)$
(E3)	$S(KI)$	$=$	I
(E4)	$[x]\,S(Kx)I$	$=$	I
(E5)	$[x,y]\,K(xy)$	$=$	$[x,y]\,S(Kx)(Ky)$

Table 2: The axioms of extensionality.

The variables appearing in the notation are used only to make explicit the reduction properties of the terms involved. Note also that each of the axioms is only a schema; proper axioms are obtained by an assignment of type to the variables and subscripts to the combinators. Given such an assignment, the terms of the equations take some type. Henceforth, the system comprising the axioms (I_σ), ($\mathrm{K}_{\sigma,\sigma'}$), ($\mathrm{S}_{\sigma,\sigma',\sigma''}$), (E1)-(E5) will be called $\mathrm{CL}_{\mathrm{ext}}$.

Theorem 2.4. *(Extensionality)*
If $tx = t'x$ is derivable in CL_{ext}, then $t = t'$ is derivable in this system, provided x does not occur in t or t'.

Proof. See [S67]. □

Inspection of the proof of Theorem 2.4 shows that it holds in fact for any extension of $\mathrm{CL}_{\mathrm{ext}}$ with closed axioms, i.e. equations containing no variables. In particular, it holds for the extensions described in the next sections.

We end this section with the introduction of two special combinators which we shall use frequently in the sequel.

Definition 2.5. For $\sigma,\sigma',\sigma'' \in \mathcal{T}(\mathcal{B})$, define

(i) $B_{\sigma,\sigma',\sigma''} \triangleq [x^{\sigma\rightarrow\sigma'}, y^{\sigma''\rightarrow\sigma}, z^{\sigma''}]\,x(yz)$

(ii) $C_{\sigma,\sigma',\sigma''} \triangleq [x^{\sigma\rightarrow(\sigma'\rightarrow\sigma'')}, y^{\sigma'}, z^{\sigma}]\,xzy$

Observe that

(i) $B_{\sigma,\sigma'\sigma''} : (\sigma \to \sigma') \to ((\sigma'' \to \sigma) \to (\sigma'' \to \sigma'))$

(ii) $C_{\sigma,\sigma',\sigma''} : (\sigma \to (\sigma' \to \sigma'')) \to (\sigma' \to (\sigma \to \sigma''))$

Going back to [CF58], $B_{\sigma,\sigma',\sigma''}$ and $C_{\sigma,\sigma',\sigma''}$ are called *elementary compositor* and *elementary permutator*, respectively. B, C, K and I interact as follows:

Proposition 2.6. The following schemata are derivable in any extension of $\mathrm{CL_{ext}}$ with closed axioms:

(a) $BCC = I$

(b) $BC(BK) = K$

Proof. By repeated application of Theorem 2.4. We prove (b). Observe that

$$BC(BK)xyz = C(BKx)yz = BKxzy = K(xz)y = xz = Kxyz.$$

Thus $BC(BK) = K$. □

3 Combinatory process algebra

Given a $\mathcal{T}(\mathcal{B})$-typed signature $\Sigma = (\mathcal{B}, \mathcal{F})$, *combinatory process algebra* (over Σ) is, as is usual, a family of sets together with a collection of operators on these sets, and is axiomatized by an equational theory extending extensional combinatory logic. The family of sets is the type structure $\mathcal{T}(\mathcal{B})$. The collection of operators consists of the combinators, $\mathcal{F}$ and,

(i) for process valued types,

- the binary operators $+$ (alternative composition), $\cdot$ (sequential composition), $||$, $\lfloor\!\lfloor$ and $|$ (parallel merge, left merge and communication merge), and * (iteration);
- the unary operators Σ (for arbitrary sums), ∂_H (encapsulation) and τ_I (hiding);

(ii) for atom valued types, the constants δ (deadlock) and τ (silent step).

We refer to [BW90] for a detailed explanation of these operators except * for which we refer to [BBP93]. The precise basic signature of combinatory process algebra is given in Table 3. Here, $\mathcal{B} = (\{A^c, A, P, D\}, A^c \subseteq A \subseteq P)$ and

$$\mathcal{T}(\mathcal{B})_\sigma = \{\sigma_1 \to (\cdots(\sigma_n \to \sigma))\cdots) \mid \sigma_i \in \mathcal{T}(\mathcal{B})\}.$$

$+_\sigma, \cdot_\sigma, \|\|_\sigma, \lfloor\!\lfloor_\sigma, \|_\sigma, {}^*{}_\sigma$	:	$\sigma \to (\sigma \to \sigma)$	for all $\sigma \in \mathcal{T}(\mathcal{B})_P$
$\Sigma_{\sigma,\sigma'}$	:	$(\sigma \to \sigma') \to \sigma'$	for all $\sigma' \in \mathcal{T}(\mathcal{B})_P$
$\partial_{H,\sigma}, \tau_{I,\sigma}$	:	$\sigma \to \sigma$	for all $\sigma \in \mathcal{T}(\mathcal{B})_P$, all $H, I \subseteq \bigcup_{\sigma \in \mathcal{T}(\mathcal{B})_{A^c}} \mathcal{T}(\Sigma)_\sigma$
$\delta_\sigma, \tau_\sigma$	:	σ	for all $\sigma \in \mathcal{T}(\mathcal{B})_A$

Table 3: The basic signature of combinatory process algebra.

We shall use the following notational conventions.

(a) Binary operators will be written infix.

(b) Opposed to the usual convention in process algebra, we will not omit the operator $\cdot$. That is, we write $t \cdot t'$ for sequential composition, and tt' for function application. Moreover, we take function application to be most binding.

To give an example of a process expression within this framework, we consider the following informal specification of a one-element buffer, that buffers elements of some data set D:

$$Buffer = (\Sigma_{d \in D}\, r(d) \cdot s(d))^* \delta \ .$$

In combinatory process algebra, this buffer has the following formal description

$$Buffer = \Sigma(r \cdot s)^* \delta$$

where we assume $r, s : D \to A^c$ and *Buffer*: P.

The axioms of combinatory process algebra, in addition to those of extensional combinatory logic, are divided into five groups. The first group, listed in Table 4, consists of the axioms of argumentwise evaluation. These axioms are self-explanatory except for, perhaps, $\mathrm{AE}_{\Sigma,\sigma,\sigma' \to \sigma''}$, which defines

$\mathrm{AE}_\square$	$[x,y,z]\,(x \square y)z$	$=$	$[x,y,z]\,(xz \square yz)$	for $\square \in \{+, \cdot, \Vert, \lfloor\!\lfloor, \mid, {}^*\}$
AE_Σ	$[x,y]\,\Sigma xy$	$=$	$[x,y]\,\Sigma(Cxy)$	
$\mathrm{AE}_\square$	$[x,y]\,\square xy$	$=$	$[x,y]\,\square(xy)$	for $\square \in \{\partial_H, \tau_I\}$
$\mathrm{AE}_\square$	$\square$	$=$	$[x]\,\square$	for $\square \in \{\delta, \tau\}$

Table 4: The axioms of argumentwise evaluation.

evaluation to be argumentwise for arbitrary sums. To give an intuitive explanation of this axiom, we consider the instance were we let r be a binary function from D to P and $d : D$. Here $\Sigma_{D,D \to P} rd$, the left-hand term, denotes $(rd_1 + rd_2 + \ldots)d$, where we let $d_1, d_2, \ldots$ range over D. The term on the right, $\Sigma_{D,P}(Crd)$, however denotes $Crdd_1 + Crdd_2 + \ldots$, which after application of the permutator C reduces to $rd_1d + rd_2d + \ldots$.
In extensional combinatory logic one can derive from the axioms of argumentwise evaluation the following distribution schemata for so-called *deferred* cancellators, permutators and compositors.

Definition 3.1. For $\diamond \in \{K, C, B\}$ and $n \in \mathbb{N}$ we define $\diamond_n$ recursively by:

(i) $\diamond_0 \equiv \diamond$,

(ii) $\diamond_{n+1} \equiv B\diamond_n$.

With this definition, taken from [CF58], we have e.g. that $K_0 \equiv K$, $K_1 \equiv BK_0 \equiv BK$, $K_2 \equiv BK_1 \equiv B(BK)$, $\cdots$

Proposition 3.2. Let $\diamond \in \{K, C, B\}$. Then the following schemata are derivable in any extension of $\mathrm{CL_{ext}}$ and the axioms of argumentwise evaluation with closed axioms:

(a) for $\square \in \{+, \cdot, \Vert, \lfloor\!\lfloor, \mid, {}^*\}$, $\diamond_n(x \square y) = (\diamond_n x) \square (\diamond_n y)$;

(b) $\diamond_n(\Sigma x) = \Sigma(\diamond_{n+1} x)$

(c) for $\square \in \{\partial_H, \tau_I\}$, $\diamond_n(\square x) = \square(\diamond_n x)$;

(d) for $\square \in \{\delta, \tau\}$, $\diamond_n \square = \square$.

Proof. By induction on n employing repeatedly Theorem 2.4. By way of example we shall prove (b) for $\diamond \equiv K$. For $n = 0$ observe that

$$\begin{array}{rcll} K_0(\Sigma x)y = K(\Sigma x)y = \Sigma x = \Sigma(Kxy) & = & \Sigma(BC(BK)xy) & \text{by 2.6(b)} \\ & = & \Sigma(C(BKx)y) & \\ & = & \Sigma(BKx)y = \Sigma(K_1x)y & \text{by } \mathrm{AE}_\Sigma. \end{array}$$

Hence $K_0(\Sigma x) = \Sigma(K_1x)$. For the induction step we first prove (†) $K_{n+1}(Cxy) = C(K_{n+2}x)y$:

$$K_{n+1}(Cxy)z = K_n(Cxyz) = K_n(xzy) = K_{n+1}(xz)y = K_{n+2}xzy = C(K_{n+2}x)yz.$$

So (†) holds. We now have that

$$\begin{array}{rcll} K_{n+1}(\Sigma x)y = K_n(\Sigma xy) & = & K_n(\Sigma(Cxy)) & \\ & = & \Sigma(K_{n+1}(Cxy)) & \text{by IH} \\ & = & \Sigma(C(K_{n+2}x)y) = \Sigma(K_{n+2}x)y & \text{by } \dagger. \end{array}$$

Thus $K_{n+1}(\Sigma x) = \Sigma(K_{n+2}x)$. □

The second group of axioms consists of the ACP_τ-axioms introduced in [BK85] and extending the ACP-axioms of [BK84]. The schemata, from which the axioms can be obtained by a type assignment to the variables and operators, are listed in Table 5. They differ from all the other schemata in so far as only restricted type assignment is permitted. That is, the labels of these schemata all carry a subscript which refers to the kind of the types of the abstracted variables in that order. To be more precise, in a schema of the form

$$\mathrm{L}_{A,P} \quad [x,y]\,\phi = [x,y]\,\psi,$$

x is assumed to be of atom-valued type whereas y is assumed to be of process-valued type. For example, a properly typed instance of $\mathrm{CM5}_{A,P,A}$, with all its type super- and subscripts shown, is e.g.

$$[x^A, y^P, z^A]\,(x^A \cdot_P y^P) \mid_P z^A = [x^A, y^P, z^A]\,(x^A \mid_P z^A) \cdot_P y^P.$$

Likewise,

$$[x^{D\to A}, y^{D\to P}]\, x^{D\to A} \mathbin{⌊\!⌊}_{D\to P} y^{D\to P} = [x^{D\to A}, y^{D\to P}]\, x^{D\to A} \cdot_{D\to P} y^{D\to P}$$

is a correctly typed instance of $\mathrm{CM2}_{A,P}$.

The third group of axiom schemata comprises the *Single Exit Iteration* axioms introduced in [BBP93][2]. The schemata, listed in Table 6, give the basic interaction properties of the process valued object t^*t' that chooses between t and t', and upon termination of t has this choice again.

The fourth group of axiom schemata, listed in Table 7, defines the operator Σ. Recalling that Σx denotes $xd_1 + xd_2 + \ldots$, the first schema, $\Sigma + \Sigma$, is obvious. The remaining schemata are versions of A3, A4, CM4, CM8, CM9, D3 and TI3, where binary sums are replaced by arbitrary ones. There are three derived schemata for so-called *powers* of Σ, which will prove useful in the sequel.

Definition 3.3. For $\diamond \in \{\Sigma, K\}$ and $n \in \mathbb{N}$ we define $\diamond^n$ recursively by:

(i) $\diamond^0 \equiv I$,

(ii) $\diamond^{n+1} \equiv B\diamond\diamond^n$.

This definition is again taken from [CF58]. For Σ, we have e.g. that $\Sigma^0 \equiv I$, $\Sigma^1 \equiv B\Sigma\Sigma^0 \equiv B\Sigma I$, $\Sigma^2 \equiv B\Sigma\Sigma^1 \equiv B\Sigma(B\Sigma I)$, $\cdots$

Proposition 3.4. The following schemata are derivable in any extension of $\mathrm{CL}_{\mathrm{ext}}$ and the Σ-axioms with closed axioms:

[2]It should be noted that we deviate from the original numbering of the SEI-axioms: the axioms SEI4 and SEI5 of Table 6 correspond in fact to the axioms SEI5 and SEI6 in [BBP93]; we omit SEI4 of [BBP93], since it can be proved in BPA_δ + SEI3.

$(\mathrm{A1}_{P,P})$	$[x,y]\,x+y$	$=$	$[x,y]\,y+x$
$(\mathrm{A2}_{P,P,P})$	$[x,y,z]\,x+(y+z)$	$=$	$[x,y,z]\,(x+y)+z$
$(\mathrm{A3}_{P})$	$[x]\,x+x$	$=$	I
$(\mathrm{A4}_{P,P,P})$	$[x,y,z]\,(x+y)\cdot z$	$=$	$[x,y,z]\,(x\cdot z)+(y\cdot z)$
$(\mathrm{A5}_{P,P,P})$	$[x,y,z]\,(x\cdot y)\cdot z$	$=$	$[x,y,z]\,x\cdot(y\cdot z)$
$(\mathrm{A6}_{P})$	$[x]\,x+\delta$	$=$	I
$(\mathrm{A7}_{P})$	$[x]\,\delta\cdot x$	$=$	δ
$(\mathrm{CM1}_{P,P})$	$[x,y]\,x\parallel y$	$=$	$[x,y,z]\,(x \mathbin{\|\!\!\lfloor} y)+(y \mathbin{\|\!\!\lfloor} x)+(x\mid y)$
$(\mathrm{CM2}_{A,P})$	$[x,y]\,x \mathbin{\|\!\!\lfloor} y$	$=$	$[x,y]\,x\cdot y$
$(\mathrm{CM3}_{A,P,P})$	$[x,y,z]\,(x\cdot y) \mathbin{\|\!\!\lfloor} z$	$=$	$[x,y,z]\,x\cdot(y\parallel z)$
$(\mathrm{CM4}_{P,P,P})$	$[x,y,z]\,(x+y) \mathbin{\|\!\!\lfloor} z$	$=$	$[x,y,z]\,(x \mathbin{\|\!\!\lfloor} z)+(y \mathbin{\|\!\!\lfloor} z)$
$(\mathrm{CM5}_{A,P,A})$	$[x,y,z]\,(x\cdot y)\mid z$	$=$	$[x,y,z]\,(x\mid z)\cdot y$
$(\mathrm{CM6}_{A,A,P})$	$[x,y,z]\,x\mid(y\cdot z)$	$=$	$[x,y,z]\,(x\mid y)\cdot z$
$(\mathrm{CM7}_{A,P,A,P})$	$[x,y,z,u]\,(x\cdot y)\mid(z\cdot u)$	$=$	$[x,y,z,u]\,(x\mid z)\cdot(y\parallel u)$
$(\mathrm{CM8}_{P,P,P})$	$[x,y,z]\,(x+y)\mid z$	$=$	$[x,y,z]\,(x\mid z)+(y\mid z)$
$(\mathrm{CM9}_{P,P,P})$	$[x,y,z]\,x\mid(y+z)$	$=$	$[x,y,z]\,(x\mid y)+(x\mid z)$
$(\mathrm{T1}_{P})$	$[x]\,x\cdot\tau$	$=$	I
$(\mathrm{T2}_{P})$	$[x]\,\tau\cdot x$	$=$	$[x]\,(\tau\cdot x)+x$
$(\mathrm{T3}_{A,P,P})$	$[x,y,z]\,x\cdot((\tau\cdot y)+z)$	$=$	$[x,y,z]\,(x\cdot((\tau\cdot y)+z))+(x\cdot y)$
$(\mathrm{TC1}_{P})$	$[x]\,\tau\mid x$	$=$	δ
$(\mathrm{TC2}_{P})$	$[x]\,x\mid\tau$	$=$	δ
$(\mathrm{TC3}_{P,P})$	$[x,y]\,(\tau\cdot x)\mid y$	$=$	$[x,y]\,x\mid y$
$(\mathrm{TC4}_{P,P})$	$[x,y]\,x\mid(\tau\cdot y)$	$=$	$[x,y]\,x\mid y$
(D0)	$\partial_H\Box$	$=$	$\Box$ for $\Box\in\{\delta,\tau\}$
(D1)	$\partial_H a$	$=$	a for $a\in\mathcal{F}-H$
(D2)	$\partial_H a$	$=$	δ for $a\in\mathcal{F}\cap H$
$(\mathrm{D3}_{P,P})$	$[x,y]\,\partial_H(x+y)$	$=$	$[x,y]\,\partial_H x+\partial_H y$
$(\mathrm{D4}_{P,P})$	$[x,y]\,\partial_H(x\cdot y)$	$=$	$[x,y]\,\partial_H x\cdot\partial_H y$
(TI0)	$\tau_I\Box$	$=$	$\Box$ for $\Box\in\{\delta,\tau\}$
(TI1)	$\tau_I a$	$=$	a for $a\in\mathcal{F}-I$
(TI2)	$\tau_I a$	$=$	τ for $a\in\mathcal{F}\cap I$
$(\mathrm{TI3}_{P,P})$	$[x,y]\,\tau_I(x+y)$	$=$	$[x,y]\,\tau_I x+\tau_I y$
$(\mathrm{TI4}_{P,P})$	$[x,y]\,\tau_I(x\cdot y)$	$=$	$[x,y]\,\tau_I x\cdot\tau_I y$

Table 5: The ACP_τ-axioms of combinatory process algebra.

(SEI1)	$[x,y]\, x\cdot(x^*y)+y$	$=$	$[x,y]\, x^*y$
(SEI2)	$[x,y,z]\, x^*(y\cdot z)$	$=$	$[x,y,z]\, (x^*y)\cdot z$
(SEI3)	$[x,y,z]\, x^*(y\cdot((x+y)^*z)+z)$	$=$	$[x,y,z]\, (x+y)^*z$
(SEI4)	$[x,y]\, \partial_H(x^*y)$	$=$	$[x,y]\, \partial_H(x)^*\partial_H(y)$
(SEI5)	$[x,y]\, \tau_I(x^*y)$	$=$	$[x,y]\, \tau_I(x)^*\tau_I(y)$

Table 6: The SEI-axioms of combinatory process algebra.

$(\Sigma+\Sigma)$	$[x,y]\,(\Sigma x)+(\Sigma y)$	$=$	$[x,y]\,\Sigma(x+y)$	
(ΣK)	$[x]\,\Sigma(Kx)$	$=$	I	
$(K\Sigma)$	$[x]\,x+K(\Sigma x)$	$=$	$[x]\,K(\Sigma x)$	
$(\Sigma\Box)$	$[x,y]\,(\Sigma x)\Box y$	$=$	$[x,y]\,\Sigma(x\Box(Ky))$	for $\Box\in\{\cdot,\ \|\!\lfloor\ ,\ \mid\}$
$(\mid\Sigma)$	$[x,y]\,x\mid(\Sigma y)$	$=$	$[x,y]\,\Sigma(Kx\mid y)$	
$(\Box\Sigma)$	$[x]\,\Box(\Sigma x)$	$=$	$[x]\,\Sigma(\Box x)$	for $\Box\in\{\partial_H,\tau_I\}$

Table 7: The Σ-axioms of combinatory process algebra.

(i) $K^{n+1}=BK^nK$;

(ii) for $\Box\in\{\cdot,\ \|\!\lfloor\ ,\ \mid\}$, $\Sigma^n x\Box y=\Sigma^n(x\Box(K^ny))$;

(iii) for $\Box\in\{\partial_H,\tau_I\}$, $\Box(\Sigma^n x)=\Sigma^n(\Box x)$;

(iv) for $\Box\in\{\delta,\tau\}$, $\Sigma^n\Box=\Box$.

Proof. By induction on n. We prove (i) and (ii).
(i) For $n=0$ we have that

$$K^1x=BKIx=K(Ix)=Kx=I(Kx)=BIKx=BK^0Kx.$$

Hence $K^1=BK^0K$ by Theorem 2.4. For $n>0$,

$$\begin{aligned}K^{n+1}x=BKK^nx &= K(K^nx)\\ &\overset{\text{IH}}{=} K(BK^{n-1}Kx)\\ &= K(K^{n-1}(Kx))=BKK^{n-1}(Kx)=K^n(Kx)=BK^nKx.\end{aligned}$$

Thus $K^{n+1}=BK^nK$ by Theorem 2.4.
For $n=0$, (ii) is immediate. For $n>0$,

$$\begin{aligned}\Sigma^nx\Box y=(B\Sigma\Sigma^{n-1}x)\Box y &= \Sigma(\Sigma^{n-1}x)\Box y\\ &= \Sigma((\Sigma^{n-1}x)\Box(Ky)) && \text{by } \Sigma\Box\\ &= \Sigma(\Sigma^{n-1}(x\Box(K^{n-1}(Ky)))) && \text{by IH}\\ &= \Sigma(\Sigma^{n-1}(x\Box(BK^{n-1}Ky)))\\ &= \Sigma(\Sigma^{n-1}(x\Box(K^ny))) && \text{by (i)}\\ &= B\Sigma\Sigma^{n-1}(x\Box(K^ny))=\Sigma^n(x\Box(K^ny))\end{aligned}$$

$\Box$

The last group of schemata, listed in Table 8, defines communication merge for atom-valued elements of $\mathcal{F}$. The first and second of these schemata state that for a, b of equal arity $ax_1 \ldots x_n \mid by_1 \ldots y_n = \delta$, if for some $1 \leq i \leq n$, $x_i \neq y_i$. Here a distinction is made between compatible and incompatible a, b. The third schema deals with a, b of different arity, say n and m. In that case $ax_1 \ldots x_n \mid by_1 \ldots y_m = \delta$. Note that these schemata are inspired by the rule CF2′ and the axiom schema CF2″ of μCRL in [GP91]. Observe also that the cases without x or y can be derived by appropriate substitution of

$(\vert_{a,b,n})$	$[x,y]\,(a \cdot x) \mid K^n(\Sigma^n(b \cdot y))$	$=$	$[x,y]\,(a \mid b) \cdot (x \parallel y)$	for compatible a, b
$(\vert'_{a,b,n})$	$[x,y]\,(a \cdot x) \mid K^n(\Sigma^n(b \cdot y))$	$=$	δ	for incompatible a, b
$(\vert_{a,b,n,m})$	$[x,y]\,(a \cdot x) \mid K^n(\Sigma^m(b \cdot y))$	$=$	δ	for $n \neq m$

Table 8: The $|_{\mathcal{F}}$-axioms of combinatory process algebra where $a, b \in \mathcal{F} \cap \mathcal{T}(\mathcal{B})_A$.

τ. The commuted cases are derivable using $[x,y]\, x \mid y = [x,y]\, y \mid x$, one of the axioms of standard concurrency that will be added lateron.

Proposition 3.5. The following schemata are derivable in any extension of $\mathrm{CL_{ext}}$, the Σ-axioms and the $|_{\mathcal{F}}$-axioms with closed axioms: for $a, b \in \mathcal{F} \cap \mathcal{T}(\mathcal{B})_A$ and $n, m \in \mathbb{N}$,

(i) $\Sigma^n(a \cdot x) \mid \Sigma^n(b \cdot y) = \Sigma^n((a \mid b) \cdot (x \parallel y))$, provided a, b are compatible;

(ii) $\Sigma^n(a \cdot x) \mid \Sigma^m(b \cdot y) = \delta$, provided $n \neq m$.

Proof. (i) and (ii) follow from Proposition 3.4:
(i) $\Sigma^n(a \cdot x) \mid \Sigma^n(b \cdot y) \overset{3.4(\mathrm{ii})}{=} \Sigma^n((a \cdot x) \mid K^n(\Sigma^n(b \cdot y))) \overset{|_{a,b,n}}{=} \Sigma^n((a \mid b) \cdot (x \parallel y))$
(ii) $\Sigma^n(a \cdot x) \mid \Sigma^m(b \cdot y) \overset{3.4(\mathrm{ii})}{=} \Sigma^n((a \cdot x) \mid K^n(\Sigma^m(b \cdot y))) \overset{|_{a,b,n,m}}{=} \Sigma^n\delta \overset{3.4(\mathrm{iv})}{=} \delta$ □

There are three additional consequences of the axiom schemata above which we shall employ in the next section.

Proposition 3.6. The following schemata are derivable in any extension of $\mathrm{CL_{ext}}$, the Σ-axioms and the $|_{\mathcal{F}}$-axioms with closed axioms: for $a, b \in \mathcal{F} \cap \mathcal{T}(\mathcal{B})_A$ and $l, m, n \in \mathbb{N}$,

(i) $BCK^{n+2} = K^{n+2}$,

(ii) $(K^l a \cdot x) \mid K^{n+l}(\Sigma^m(b \cdot y)) = \delta$, provided $n \neq m$;

(iii) $(BKa \cdot x) \mid K^{n+2}(\Sigma^m(b \cdot y)) = \delta$, provided $n + 1 \neq m$;

(iv) $Ka \mid BKb = \delta$, provided a, b are incompatible.

Proof. For (i) observe that

$$BCK^{n+2}xyz = C(K^{n+2}x)yz = K^{n+2}xzy = K^n x = K^{n+2}xyz.$$

For (ii) note that

$$\begin{aligned} ((K^l a \cdot x) \mid K^{n+l}(\Sigma^m(b \cdot y)))z_1 \ldots z_l &= (K^l a \cdot x)z_1 \ldots z_l \mid K^n(\Sigma^m(b \cdot y) && \text{by AE}_{|} \\ &= (a \cdot xz_1 \ldots z_l) \mid K^n(\Sigma^m(b \cdot y)) && \text{by AE.} \\ &= \delta = K^l \delta z_1 \ldots z_l && \text{by } |_{a,b,n,m}\,. \end{aligned}$$

Thus $(K^l a \cdot x) \mid K^{n+l}(\Sigma^m(b \cdot y)) = K^l \delta \stackrel{\mathrm{AE}_\delta}{=} \delta$.
(iii)

$$\begin{array}{rcll}
(BKa \cdot x) \mid K^{n+2}(\Sigma^m(b \cdot y)) & = & BCC((BKa \cdot x) \mid K^{n+2}(\Sigma^m(b \cdot y))) & \text{by 2.6(a)} \\
& = & C(C((BKa \cdot x) \mid K^{n+2}(\Sigma^m(b \cdot y)))) & \\
& = & C((C(BKa) \cdot Cx) \mid C(K^{n+2}(\Sigma^m(b \cdot y)))) & \text{by 3.2(a)} \\
& = & C((BC(BK)a \cdot Cx) \mid BCK^{n+2}(\Sigma^m(b \cdot y))) & \\
& = & C((Ka \cdot Cx) \mid K^{n+2}(\Sigma^m(b \cdot y))) & \text{by 2.6(b), (i)} \\
& = & C\delta & \text{by (ii)} \\
& = & \delta & \text{by 3.2(d)}
\end{array}$$

(iv) First observe that

$$(\dagger) \quad \Sigma(Ka \mid BKb) \stackrel{|\Sigma}{=} a \mid \Sigma(BKb) \stackrel{3.2(b)}{=} a \mid K(\Sigma b) \stackrel{\mathrm{T1},|'_{a,b,1}}{=} \delta.$$

Hence

$$\begin{array}{rcll}
Ka \mid BKb & = & (Ka \mid BKb) + \delta & \text{by A6} \\
& = & (Ka \mid BKb) + K\delta & \text{by AE}_\delta \\
& = & (Ka \mid BKb) + K(\Sigma(Ka \mid BKb)) & \text{by } \dagger \\
& = & K(\Sigma(Ka \mid BKb)) & \text{by } K\Sigma \\
& = & K\delta & \text{by } \dagger \\
& = & \delta & \text{by AE}_\delta.
\end{array}$$

□

This ends the description of the formal system of combinatory process algebra.

A *combinatory process specification* consists of a $\mathcal{T}(\mathcal{B})$-typed signature, Σ, together with a set $\mathcal{E}$ of combinatory process algebra equations over Σ. We shall write

$$\mathcal{E} \vdash t = t',$$

if $t = t'$ is derivable from $\mathcal{E}$ and the axioms listed in Table 1, 2, 4, 5, 6, 7 and 8 by means of equational logic.

4 A working example: the Simple Alternating Bit Protocol

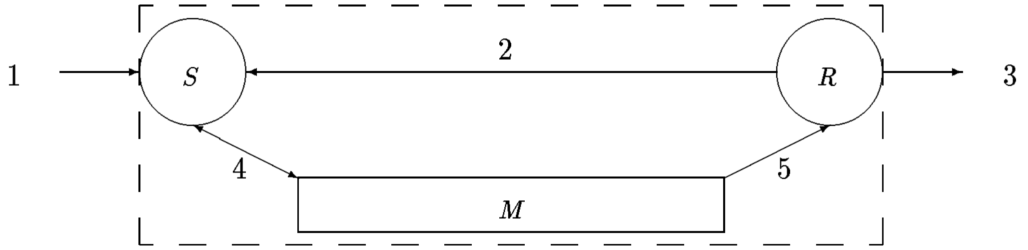

Figure 1: The Simple Alternating Bit Protocol

The Simple Alternating Bit Protocol (SABP), essentially due to [P85] and reconsidered in [BKO87], is an idealized communication protocol providing reliable transmission of data through an unreliable medium. Externally, the behaviour of the protocol is that of a buffer reading data from some input and subsequently writing the data to some output. Its internal behaviour, however, is determined by a more complex interaction between a sender S, a medium M and a receiver R, which are connected via directed communication channels as depicted in Figure 1. The sender inputs data from channel 1 and forwards frames consisting of a datum and a bit into the medium via channel 4. These actions are represented by $r_1 : D \to A^c$ and $s_4 : D \to (\mathbb{B} \to A^c)$. The medium forwards frames to the receiver via channel 5 or sends an error indication, representing loss or corruption of a datum, to the sender along channel 4. These actions are represented by $s_5 : D \to (\mathbb{B} \to A^c)$ and $s_\perp : A^c$. The receiver writes data to the output via channel 3 and acknowledges receipt along channel 2. These actions are represented by $s_3 : D \to A^c$ and $s_2 : \mathbb{B} \to A^c$. A read or send action in a component along a certain channel has a send or read counterpart in the component with which the channel in question is shared. Communication is synchronous, i.e. only occurs when complementary r/s actions are executed simultaneously through the same channel. The resulting actions are denoted by $c_\perp, c_2, c_4, c_5$. The SABP roughly works as follows: S reads a datum d from the input and sends a frame (b, d) via M to R. If S gets the acknowledgement bit b from the receiver, it can input the next datum d' and forward it together with the switched bit b' along channel 4. If, however, S receives an error indication $\perp$ from the medium, S retransmits the present frame before inputting a new one. It should be clear that the alternating bit is essential to distinguish new frames from old ones.

$0, 1$	:	$\mathbb{B}$
$i, r_\perp, s_\perp, c_\perp$	:	A^c
r_1, s_3	:	$D \to A^c$
r_2, s_2, c_2	:	$\mathbb{B} \to A^c$
$r_4, s_4, c_4, r_5, s_5, c_5$	:	$D \to (\mathbb{B} \to A^c)$
$S, M, R, SABP$	:	P
G	:	$\mathbb{B} \to P$
T	:	$D \to (\mathbb{B} \to P)$

Table 9: The function symbols of SABP.

The specification of SABP in combinatory process algebra consists of the signature Σ_{SABP} where $\mathcal{B}_{\text{SABP}} = (\{\mathbb{B}, A^c, A, D, P\}, A^c \subseteq A \subseteq P)$ and $\mathcal{F}_{\text{SABP}}$ is listed in Table 9, together with the axioms $\mathcal{E}_{\text{SABP}}$ listed in Table 10. Observe that we prefix the send actions of the medium with an atomic action i in order to make the choice non-deterministic: that is, the decision whether or not a frame will be corrupted is internal to the medium, and cannot be influenced by the environment.

The question is now whether SABP is specified correctly: does the entire process, apart from its internal actions, behave as a one-element buffer? Or, to put it differently, can we prove that

$$\tau_{I,P}(SABP) = \Sigma_{D,P}(r_1 \cdot s_3)^* \delta$$

where $I = \{c_\perp, c_2, c_4, c_5, i\}$? As is usual (cf. e.g. [BK86], [V90], [BG93]), we assume that the following principles are satisfied in the algebra in which we model SABP: *Standard Concurrency* (SC), *Fair Abstraction* and the *Recursive Specification Principle* (RSP). The latter principle may be problematic but certainly not in the case of finite data types.

Standard Concurrency is an extension of process algebra originally due to [BT84]. We list its axiom schemata in Table 11. All axioms SC1-SC6 hold in fact for finite processes from ACP_τ. In [BW90] these axioms are proved with induction on term formation. We will not need here SC2 and SC4.

Fair Abstraction is the principle that certain abstracted process steps will be fairly scheduled in such a way that eventually a unabstracted step is performed. In the case of weak bisimulation semantics, the

(M)	M	$=$	$\Sigma_{\mathbb{B},P}(\Sigma_{D,\mathbb{B}\to P}(r_4 \cdot (\, K(Ki) \cdot s_5 \,+\, K(K(i \cdot s_\perp)))))^* \delta$
(T)	T	$=$	$(K(Kr_\perp) \cdot s_4)^*(Kr_2)$
(G)	G	$=$	$\Sigma_{D,\mathbb{B}\to P}(BKr_1 \cdot s_4 \cdot T)$
(S)	S	$=$	$(G0 \cdot G1)^* \delta$
(R)	R	$=$	$\Sigma_{\mathbb{B},P}(\Sigma_{D,\mathbb{B}\to P}(r_5 \cdot BKs_3 \cdot Ks_2))^* \delta$
$(\vert_c)$	$s_j \mid r_j$	$=$	c_j for $j \in \{\perp, 2, 4, 5\}$
$(\vert_\delta)$	$a \mid a'$	$=$	δ for all non $-$ complementary actions $a, a' \in \mathcal{F}$
(SABP)	$SABP$	$=$	$\partial_{H,P}((S \parallel M) \parallel R)$ for $H = \{r_\perp, s_\perp, r_2, s_2, r_4, s_4, r_5, s_5\}$

Table 10: The axioms of SABP.

(SC1)	$[x,y,z]\,(x \mathbin{\|\!\!\lfloor} y) \mathbin{\|\!\!\lfloor} z$	$=$	$[x,y,z]\,x \mathbin{\|\!\!\lfloor} (y \parallel z)$
(SC2)	$[x]\,x \mathbin{\|\!\!\lfloor} \delta$	$=$	$[x]\,x \cdot \delta$
(SC3)	$[x,y]\,x \mid y$	$=$	$[x,y]\,y \mid x$
(SC4)	$[x,y,z]\,(x \mid y) \mid z$	$=$	$[x,y,z]\,x \mid (y \mid z)$
(SC5)	$[x,y,z]\,x \mid (y \mathbin{\|\!\!\lfloor} z)$	$=$	$[x,y,z]\,(x \mid y) \mathbin{\|\!\!\lfloor} z$
(SC6)	$[x,y,z]\,x \parallel (y \parallel z)$	$=$	$[x,y,z]\,(x \parallel y) \parallel z$

Table 11: The axioms of Standard Concurrency (SC).

principle is guaranteed by *Koomen's Fair Abstraction Rules* KFAR_n, introduced in [BK84]. KFAR_1, which is sufficient in the case of SABP, reads as follows:

$$\frac{x = ix + y \quad (i \in I)}{\tau_I(x) = \tau \cdot \tau_I(y)}$$

(so the infinite τ sequence induced by ix is reduced to a single τ step). In the presence of *, however, we can replace this rule by the Fair Iteration Axiom FIR_1 of [BBP93]. The closed axiom schema corresponding to this axiom is given in Table 12.

$$(\mathrm{FIR}_1) \quad [x]\,\tau^* x \;=\; [x]\,\tau \cdot x$$

Table 12: The Fair Iteration Axiom Schema FIR_1.

The important principle RSP, introduced in [BK86], expresses the fact that each guarded recursive equation has at most one solution. In a setting without τ, a *-adaptation of RSP can be given by

$$(\mathrm{RSP}^*) \quad \frac{x = y \cdot x + z}{x = y^* z.}$$

However, in the presence of τ, the rule RSP^* as such is not sound and anyhow a too heavy tool for our purposes. Here its weaker version

$$(\mathrm{wRSP}^*) \quad \frac{x = t \cdot x + y}{x = t^* y}$$

suffices where y is replaced by a *τ-free* term, that is a closed term containing no occurrences of τ or of an operator τ_I. As we wish to stay within the framework of pure equational reasoning, we shall not adopt this rule here. Instead, we extend our signature by two new operators, $EQ : P \to (P \to \mathbb{B})$ and $\lhd\ \rhd : P \to (\mathbb{B} \to (P \to P))$, and use the wRSP^*-axioms listed in Table 13. Observe that applying

(EQ0)	$[x]\,EQxx$	$=$	$[x]\,0$
(EQ1)	$[x,y]\,EQxy$	$=$	$[x,y]\,EQyx$
$(\lhd\ \rhd 0)$	$[x,y]\,x \lhd 0 \rhd y$	$=$	$[x,y]\,x$
$(\lhd\ \rhd 1)$	$[x,y]\,x \lhd 1 \rhd y$	$=$	$[x,y]\,y$
$(\lhd\ \rhd 2)$	$[x,y]\,x \lhd (EQxy) \rhd y$	$=$	$[x,y]\,y$
$(\lhd\ \rhd 3)$	$[x,y]\,(t^*y) \lhd (EQx(t \cdot x + y)) \rhd x$	$=$	$[x,y]\,x$

Table 13: The axioms for the Weak Recursive Specification Principle where t is τ-free.

these axioms indeed yields

Proposition 4.1. Let $\mathrm{wRSP}^* \subseteq \mathcal{E}$, and let t be a τ-free term and t', t'' be arbitrary (open or closed) terms. If $\mathcal{E} \vdash t'' = t \cdot t'' + t'$, then $\mathcal{E} \vdash t'' = t^* t'$.

Proof. Assume (†) $\quad t'' = t \cdot t'' + t'$. Then

$$\begin{array}{rcll}
t'' & = & (t^*t') \lhd (EQt''(t \cdot t'' + t')) \rhd t'' & \text{by } \lhd\ \rhd 3 \\
 & = & (t^*t') \lhd (EQt''t'') \rhd t'' & \text{by } \dagger \\
 & = & (t^*t') \lhd 0 \rhd t'' & \text{by EQ0} \\
 & = & t^*t' & \text{by } \lhd\ \rhd 0
\end{array}$$

□

It is thus rather the following consequence of which we shall verify the proof:

$$\mathcal{E}_{\text{SABP}} \cup \text{SC} \cup \text{FIR}_1 \cup \text{wRSP}^* \vdash \tau_{I,P}(SABP) = \Sigma_{D,P}(r_1 \cdot s_3)^*\delta.$$

The proof is headed by five linearization steps, the propositions 4.3, 4.4, 4.6, 4.8 and 4.10, each of which, except for Proposition 4.4, is preceded by a lemma comprising a few subcalculations. The preparatory steps are combined in Theorem 4.11. We present all calculations rather detailed - almost finicky - in order to illustrate thoroughly the application of the axioms of combinatory process algebra.

The linearization of the specification of SABP constitutes the main part of the proof. For the first step, the following equalities are needed.

Lemma 4.2.

(i) $\mathcal{E}_{\text{SABP}} \vdash Gx \cdot y = \Sigma(r_1 \cdot C(s_4 \cdot T)x \cdot Ky)$

(ii) $\mathcal{E}_{\text{SABP}} \vdash M = \Sigma^2(r_4 \cdot (\, K^2 i \cdot s_5 \;+\; K^2(i \cdot s_\perp)\,) \cdot K^2 M)$

(iii) $\mathcal{E}_{\text{SABP}} \vdash R = \Sigma^2(r_5 \cdot BKs_3 \cdot Ks_2 \cdot K^2 R)$

(iv) For $t \in \{M, R\}$, $\mathcal{E}_{\text{SABP}} \vdash (Gx \cdot y) \mid t = \delta.$

(v) $\mathcal{E}_{\text{SABP}} \vdash R \mid M = \delta$

(vi) For $t \in \{M, R\}$, $\mathcal{E}_{\text{SABP}} \vdash \partial_H(M \mathbin{|\!\!\lfloor} x) = \delta.$

Proof. (i)

$$\begin{array}{rcll}
Gx \cdot y & = & \Sigma(BKr_1 \cdot s_4 \cdot T)x \cdot y & \text{by G} \\
& = & \Sigma(C(BKr_1 \cdot s_4 \cdot T)x) \cdot y & \text{by AE}_\Sigma \\
& = & \Sigma((C(BKr_1) \cdot C(s_4 \cdot T))x) \cdot y & \text{by 3.2(a)} \\
& = & \Sigma(C(BKr_1)x \cdot C(s_4 \cdot T)x) \cdot y & \text{by AE.} \\
& = & \Sigma(BC(BK)r_1 x \cdot C(s_4 \cdot T)x) \cdot y & \\
& = & \Sigma(Kr_1 x \cdot C(s_4 \cdot T)x) \cdot y & \text{by 2.6(b)} \\
& = & \Sigma(r_1 \cdot C(s_4 \cdot T)x) \cdot y & \\
& = & \Sigma(r_1 \cdot C(s_4 \cdot T)x \cdot Ky) & \text{by } \Sigma\cdot
\end{array}$$

The proofs of (ii) and (iii) follow a same pattern. We prove (ii).

$$\begin{array}{rcll}
M & = & \Sigma(\Sigma(r_4 \cdot (K(Ki) \cdot s_5 + K(K(i \cdot s_\perp))))))^*\delta & \text{by M} \\
& = & \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)))^*\delta & \\
& = & \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp))) \cdot (\Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)))^*\delta) + \delta & \text{by SEI1} \\
& = & \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp))) \cdot M + \delta & \text{by M} \\
& = & \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp))) \cdot M & \text{by A6} \\
& = & \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) & \text{by 3.4(ii)}
\end{array}$$

We prove (iv) for $t \equiv M$.

$$\begin{array}{rcll}
(Gx \cdot y) \mid M & = & \Sigma(r_1 \cdot C(s_4 \cdot T)x \cdot Ky) \mid \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) & \text{by (i), (ii)} \\
& = & \delta & \text{by 3.5(ii)}
\end{array}$$

(v)

$$\begin{array}{rcll}
R \mid M & = & \Sigma^2(r_5 \cdot BKs_3 \cdot Ks_2 \cdot K^2 R) \mid \Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) & \text{by (ii), (iii)} \\
& = & \Sigma^2((r_5 \mid r_4) \cdot ((BKs_3 \cdot Ks_2 \cdot K^2 R) \parallel ((K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M))) & \text{by 3.5(i)} \\
& = & \Sigma^2(\delta \cdot ((BKs_3 \cdot Ks_2 \cdot K^2 R) \parallel ((K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M))) & \text{by } |_\delta \\
& = & \Sigma^2\delta & \text{by A7} \\
& = & \delta & \text{by 3.4(iv)}
\end{array}$$

We prove (vi) for $t \equiv M$.

$$
\begin{array}{rcll}
\partial_H(M \mathbin{\lfloor\!\lfloor} x) & = & \partial_H(\Sigma^2(r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) \mathbin{\lfloor\!\lfloor} x) & \text{by (ii)} \\
& = & \partial_H(\Sigma^2((r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) \mathbin{\lfloor\!\lfloor} K^2 x)) & \text{by 3.4(ii)} \\
& = & \Sigma^2(\partial_H((r_4 \cdot (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) \mathbin{\lfloor\!\lfloor} K^2 x)) & \text{by 3.4(iii)} \\
& = & \Sigma^2(\partial_H(r_4 \cdot (((K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) \parallel K^2 x))) & \text{by CM3} \\
& = & \Sigma^2(\partial_H r_4 \cdot \partial_H(((K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) \parallel K^2 x)) & \text{by D4} \\
& = & \Sigma^2(\delta \cdot \partial_H(((K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M) \parallel K^2 x)) & \text{by D2} \\
& = & \Sigma^2 \delta & \text{by A7} \\
& = & \delta & \text{by 3.4(iv)}
\end{array}
$$

□

Proposition 4.3.

$$\mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \vdash \partial_H(((Gx \cdot y) \parallel M) \parallel R) = \Sigma(r_1 \cdot C(\partial_H((s_4 \cdot T \cdot K^2 y) \parallel K^2(M \parallel R)))x)$$

Proof.

$$
\begin{array}{rcll}
\partial_H(((Gx \cdot y) \parallel M) \parallel R) & = & \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M + M \mathbin{\lfloor\!\lfloor} (Gx \cdot y)) \parallel R) & \text{by CM1, 4.2(iv), A6} \\
& = & \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M + M \mathbin{\lfloor\!\lfloor} (Gx \cdot y)) \mathbin{\lfloor\!\lfloor} R) & \\
& & + \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M + M \mathbin{\lfloor\!\lfloor} (Gx \cdot y)) \mid R) & \text{by CM1, D3, 4.2(vi), A6} \\
& = & \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M) \mathbin{\lfloor\!\lfloor} R + M \mathbin{\lfloor\!\lfloor} ((Gx \cdot y) \parallel R)) & \\
& & + \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M + M \mathbin{\lfloor\!\lfloor} (Gx \cdot y)) \mid R) & \text{by CM4, SC1} \\
& = & \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M) \mathbin{\lfloor\!\lfloor} R) & \\
& & + \partial_H(((Gx \cdot y) \mathbin{\lfloor\!\lfloor} M + M \mathbin{\lfloor\!\lfloor} (Gx \cdot y)) \mid R) & \text{by D3, 4.2(vi), A6} \\
& = & \partial_H((Gx \cdot y) \mathbin{\lfloor\!\lfloor} (M \parallel R)) & \\
& & + \partial_H((R \mid M) \mathbin{\lfloor\!\lfloor} (Gx \cdot y) + ((Gx \cdot y) \mid R) \parallel M) & \text{by CM8, SC1, 3, 5} \\
& = & \partial_H((Gx \cdot y) \mathbin{\lfloor\!\lfloor} (M \parallel R)) & \text{by 4.2(iv), (v), CM2,} \\
& & & \text{A6, 7, D0} \\
& = & \partial_H(\Sigma(r_1 \cdot C(s_4 \cdot T)x \cdot Ky)) \mathbin{\lfloor\!\lfloor} (M \parallel R)) & \text{by 4.2(i)} \\
& = & \partial_H(\Sigma((r_1 \cdot C(s_4 \cdot T)x \cdot Ky) \mathbin{\lfloor\!\lfloor} K(M \parallel R))) & \text{by } \Sigma \mathbin{\lfloor\!\lfloor} \\
& = & \Sigma(\partial_H((r_1 \cdot C(s_4 \cdot T)x \cdot Ky) \mathbin{\lfloor\!\lfloor} K(M \parallel R))) & \text{by } \partial\Sigma \\
& = & \Sigma(\partial_H(r_1 \cdot ((C(s_4 \cdot T)x \cdot Ky) \parallel K(M \parallel R)))) & \text{by CM3} \\
& = & \Sigma(\partial_H r_1 \cdot \partial_H((C(s_4 \cdot T)x \cdot Ky) \parallel K(M \parallel R))) & \text{by D4} \\
& = & \Sigma(r_1 \cdot \partial_H((C(s_4 \cdot T)x \cdot Ky) \parallel K(M \parallel R))) & \text{by D1} \\
& = & \Sigma(r_1 \cdot \partial_H((C(s_4 \cdot T)x \cdot K^2 yx) \parallel K^2(M \parallel R)x)) & \\
& = & \Sigma(r_1 \cdot \partial_H(((C(s_4 \cdot T) \cdot K^2 y) \parallel K^2(M \parallel R))x)) & \text{by AE., AE}_\parallel \\
& = & \Sigma(r_1 \cdot \partial_H((C(s_4 \cdot T) \cdot K^2 y) \parallel K^2(M \parallel R))x) & \text{by AE}_\partial \\
& = & \Sigma(r_1 \cdot \partial_H((C(s_4 \cdot T) \cdot BCK^2 y) \parallel BCK^2(M \parallel R))x) & \text{by 3.6(i)} \\
& = & \Sigma(r_1 \cdot \partial_H((C(s_4 \cdot T) \cdot C(K^2 y)) \parallel C(K^2(M \parallel R)))x) & \\
& = & \Sigma(r_1 \cdot \partial_H(C((s_4 \cdot T \cdot K^2 y) \parallel K^2(M \parallel R)))x) & \text{by 3.2(a)} \\
& = & \Sigma(r_1 \cdot C(\partial_H((s_4 \cdot T \cdot K^2 y) \parallel K^2(M \parallel R)))x) & \text{by 3.2(c)}
\end{array}
$$

□

The second linearization step looks like this.

Proposition 4.4.

$$\mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \vdash \partial_H((s_4 \cdot T \cdot K^2 y) \parallel K^2(M \parallel R)) = c_4 \cdot \partial_H(((T \cdot K^2 y) \parallel ((K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M)) \parallel K^2 R)$$

Proof. Put $\Gamma \equiv (K^2 i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2 M$ and $\Gamma' \equiv (BKs_3 \cdot Ks_2 \cdot K^2 R)$. Then

$$
\begin{aligned}
\partial_H((s_4\cdot T\cdot K^2y)\parallel K^2(M\parallel R)) &= \partial_H((s_4\cdot T\cdot K^2y)\mid K^2(M \mathbin{\lfloor\!\lfloor} R+R\mathbin{\lfloor\!\lfloor} M)) && \text{by CM1, 4.2(v), A6}\\
&= \partial_H((s_4\cdot T\cdot K^2y)\mid (K^2M \mathbin{\lfloor\!\lfloor} K^2R+K^2R\mathbin{\lfloor\!\lfloor} K^2M)) && \text{by 3.2(a)}\\
&= \partial_H((s_4\cdot T\cdot K^2y)\mid (K^2M \mathbin{\lfloor\!\lfloor} K^2R)\\
&\qquad +(s_4\cdot T\cdot K^2y)\mid (K^2R \mathbin{\lfloor\!\lfloor} K^2M)) && \text{by CM9}\\
&= \partial_H(((s_4\cdot T\cdot K^2y)\mid K^2M) \mathbin{\lfloor\!\lfloor} K^2R\\
&\qquad +((s_4\cdot T\cdot K^2y)\mid K^2R) \mathbin{\lfloor\!\lfloor} K^2M) && \text{by SC5}\\
&= \partial_H(((s_4\mid r_4)\cdot((T\cdot K^2y)\parallel\Gamma)) \mathbin{\lfloor\!\lfloor} K^2R\\
&\qquad +((s_4\mid r_5)\cdot((T\cdot K^2y)\parallel\Gamma')) \mathbin{\lfloor\!\lfloor} K^2M) && \text{by 4.2(ii), (iii), } |_{s_4,r_4,2}, |_{s_4,r_5,2}\\
&= \partial_H(((s_4\mid r_4)\cdot((T\cdot K^2y)\parallel\Gamma)) \mathbin{\lfloor\!\lfloor} K^2R+\delta \mathbin{\lfloor\!\lfloor} K^2M) && \text{by } |_\delta, \text{A7}\\
&= \partial_H(((s_4\mid r_4)\cdot((T\cdot K^2y)\parallel\Gamma)) \mathbin{\lfloor\!\lfloor} K^2R) && \text{by CM2, A6, 7}\\
&= \partial_H((c_4\cdot((T\cdot K^2y)\parallel\Gamma)) \mathbin{\lfloor\!\lfloor} K^2R) && \text{by } |_c\\
&= \partial_H(c_4\cdot(((T\cdot K^2y)\parallel\Gamma) \mathbin{\lfloor\!\lfloor} K^2R)) && \text{by CM3}\\
&= \partial_H c_4\cdot\partial_H(((T\cdot K^2y)\parallel\Gamma)\parallel K^2R) && \text{by D4}\\
&= c_4\cdot\partial_H(((T\cdot K^2y)\parallel\Gamma)\parallel K^2R) && \text{by D1}
\end{aligned}
$$

□

In the third linearization step we apply the equalities below.

Lemma 4.5.

(i) $\mathcal{E}_{\mathrm{SABP}}\vdash T\cdot x=(K^2r_\perp\cdot s_4\cdot T\cdot x)+(Kr_2\cdot x)$

(ii) $\mathcal{E}_{\mathrm{SABP}}\vdash (T\cdot x)\mid(K^2i\cdot y+K^2i\cdot z)=\delta$

(iii) $\mathcal{E}_{\mathrm{SABP}}\vdash \partial_H((T\cdot x)\mathbin{\lfloor\!\lfloor} y)=\delta$

(iv) $\mathcal{E}_{\mathrm{SABP}}\vdash \partial_H((K^2i\cdot x+K^2i\cdot y)\mathbin{\lfloor\!\lfloor} z)=K^2i\cdot\partial_H(x\parallel z)+K^2i\cdot\partial_H(y\parallel z)$

(v) For $t\in\{M,R\}$, $\mathcal{E}_{\mathrm{SABP}}\vdash (T\cdot x)\mid K^2t=\delta$.

(vi) $\mathcal{E}_{\mathrm{SABP}}\vdash (K^2i\cdot x+K^2i\cdot y)\mid K^2R=\delta$

(vii) For $t\in\{M,R\}$, $\mathcal{E}_{\mathrm{SABP}}\vdash \partial_H(K^2\mathbin{\lfloor\!\lfloor} x)=\delta$.

Proof. (i)

$$
\begin{aligned}
T\cdot x &= ((K(Kr_\perp)\cdot s_4)^*(Kr_2))\cdot x && \text{by T}\\
&= ((K^2r_\perp\cdot s_4)^*(Kr_2))\cdot x\\
&= (((K^2r_\perp\cdot s_4)\cdot((K^2r_\perp\cdot s_4)^*(Kr_2)))+Kr_2)\cdot x && \text{by SEI1}\\
&= ((K^2r_\perp\cdot s_4\cdot T)+Kr_2)\cdot x && \text{by T}\\
&= (K^2r_\perp\cdot s_4\cdot T\cdot x)+(Kr_2\cdot x) && \text{by A4}
\end{aligned}
$$

(ii) We abbreviate $s_4\cdot T\cdot x$ by Γ.

$$
\begin{aligned}
(T\cdot x)\mid(K^2i\cdot y+K^2i\cdot z) &= ((K^2r_\perp\cdot\Gamma)+(Kr_2\cdot x))\mid(K^2i\cdot y+K^2i\cdot z) && \text{by (i)}\\
&= (K^2r_\perp\cdot\Gamma)\mid(K^2i\cdot y)+(K^2r_\perp\cdot\Gamma)\mid(K^2i\cdot z)\\
&\quad +(Kr_2\cdot x)\mid(K^2i\cdot y)+(Kr_2\cdot x)\mid(K^2i\cdot z) && \text{by CM8, 9}\\
&= (K^2r_\perp\mid K^2i)\cdot(\Gamma\parallel y)+(K^2r_\perp\mid K^2i)\cdot(\Gamma\parallel z)\\
&\quad +(Kr_2\mid K^2i)\cdot(x\parallel y)+(Kr_2\mid K^2i)\cdot(x\parallel z) && \text{by CM7}\\
&= K^2(r_\perp\mid i)\cdot(\Gamma\parallel y)+K^2(r_\perp\mid i)\cdot(\Gamma\parallel z)\\
&\quad +K(r_2\mid Ki)\cdot(x\parallel y)+K(r_2\mid Ki)\cdot(x\parallel z) && \text{by 3.2(a)}\\
&= K^2\delta\cdot(\Gamma\parallel y)+K^2\delta\cdot(\Gamma\parallel z)+K\delta\cdot(x\parallel y)+K\delta\cdot(x\parallel z) && \text{by } |_\delta, \text{T1, } |_{r_2,i,1,0}\\
&= \delta\cdot(\Gamma\parallel y)+\delta\cdot(\Gamma\parallel z)+\delta\cdot(x\parallel y)+\delta\cdot(x\parallel z) && \text{by 3.2(d)}\\
&= \delta && \text{by A6, 7}
\end{aligned}
$$

(iii)

$$\begin{array}{rcll}
\partial_H((T\cdot x)\mathbin{\|\!_}\, y) & = & \partial_H(((K^2r_\perp\cdot s_4\cdot T\cdot x)+(Kr_2\cdot x))\mathbin{\|\!_}\, y) & \text{by (i)}\\
& = & \partial_H((K^2r_\perp\cdot s_4\cdot T\cdot x)\mathbin{\|\!_}\, y)+\partial_H((Kr_2\cdot x)\mathbin{\|\!_}\, y) & \text{by CM4, D3}\\
& = & \partial_H(K^2r_\perp\cdot((s_4\cdot T\cdot x)\parallel y))+\partial_H(Kr_2\cdot(x\parallel y)) & \text{by CM3}\\
& = & \partial_H(K^2r_\perp)\cdot\partial_H((s_4\cdot T\cdot x)\parallel y)+\partial_H(Kr_2)\cdot\partial_H(x\parallel y) & \text{by D4}\\
& = & K^2(\partial_Hr_\perp)\cdot\partial_H((s_4\cdot T\cdot x)\parallel y)+K(\partial_Hr_2)\cdot\partial_H(x\parallel y) & \text{by 3.2(c)}\\
& = & K^2\delta\cdot\partial_H((s_4\cdot T\cdot x)\parallel y)+K\delta\cdot\partial_H(x\parallel y) & \text{by D2}\\
& = & \delta\cdot\partial_H((s_4\cdot T\cdot x)\parallel y)+\delta\cdot\partial_H(x\parallel y) & \text{by 3.2(d)}\\
& = & \delta & \text{by A6, 7}
\end{array}$$

(iv)

$$\begin{array}{rcll}
\partial_H((K^2i\cdot x+K^2i\cdot y)\mathbin{\|\!_}\, z) & = & \partial_H((K^2i\cdot x)\mathbin{\|\!_}\, z)+\partial_H((K^2i\cdot y)\mathbin{\|\!_}\, z) & \text{by CM4, D3}\\
& = & \partial_H(K^2i\cdot(x\parallel z))+\partial_H(K^2i\cdot(y\parallel z)) & \text{by CM3}\\
& = & \partial_H(K^2i)\cdot\partial_H(x\parallel z)+\partial_H(K^2i)\cdot\partial_H(y\parallel z) & \text{by D4}\\
& = & K^2(\partial_Hi)\cdot\partial_H(x\parallel z)+K^2(\partial_Hi)\cdot\partial_H(y\parallel z) & \text{by 3.2(c)}\\
& = & K^2i\cdot\partial_H(x\parallel z)+K^2i\cdot\partial_H(y\parallel z) & \text{by D1}
\end{array}$$

(v) and (vi) are proved similarly. We prove (v).

$$\begin{array}{rcll}
(T\cdot x)\mid K^2R & = & ((K^2r_\perp\cdot s_4\cdot T\cdot x)+(Kr_2\cdot x))\mid K^2(\Sigma^2(r_5\cdot BKs_3\cdot Ks_2\cdot K^2R)) & \text{by (i), 4.2(iii)}\\
& = & (K^2r_\perp\cdot s_4\cdot T\cdot x)\mid K^2(\Sigma^2(r_5\cdot BKs_3\cdot Ks_2\cdot K^2R)) & \\
& & +(Kr_2\cdot x)\mid K^2(\Sigma^2(r_5\cdot BKs_3\cdot Ks_2\cdot K^2R)) & \text{by CM8}\\
& = & \delta & \text{by 3.6(ii), A6}
\end{array}$$

We prove (vii) for $t\equiv M$. Let $\Gamma\equiv(K^2i\cdot s_5+K^2(i\cdot s_\perp))\cdot K^2M$. Then

$$\begin{array}{rcll}
\partial_H(K^2M\mathbin{\|\!_}\, x) & = & \partial_H(K^2(\Sigma^2(r_4\cdot\Gamma))\mathbin{\|\!_}\, x) & \text{by 4.2(ii)}\\
& = & \partial_H(K(K(\Sigma^2(r_4\cdot\Gamma)))\mathbin{\|\!_}\, x) & \\
& = & \partial_H(\Sigma^2(K_1(K_1(r_4\cdot\Gamma)))\mathbin{\|\!_}\, x) & \text{by 3.2(b)}\\
& = & \partial_H(\Sigma^2(K_1(K_1(r_4\cdot\Gamma))\mathbin{\|\!_}\, K^2x)) & \text{by 3.4(ii)}\\
& = & \partial_H(\Sigma^2((K_1(K_1r_4)\cdot K_1(K_1\Gamma))\mathbin{\|\!_}\, K^2x)) & \text{by 3.2(a)}\\
& = & \partial_H(\Sigma^2(K_1(K_1r_4)\cdot(K_1(K_1\Gamma)\parallel K^2x))) & \text{by CM3}\\
& = & \Sigma^2(\partial_H(K_1(K_1r_4)\cdot(K_1(K_1\Gamma)\parallel K^2x))) & \text{by 3.4(iii)}\\
& = & \Sigma^2(\partial_H(K_1(K_1r_4))\cdot\partial_H(K_1(K_1\Gamma)\parallel K^2x)) & \text{by D4}\\
& = & \Sigma^2(K_1(K_1(\partial_Hr_4))\cdot\partial_H(K_1(K_1\Gamma)\parallel K^2x)) & \text{by 3.2(c)}\\
& = & \Sigma^2(K_1(K_1\delta)\cdot\partial_H(K_1(K_1\Gamma)\parallel K^2x)) & \text{by D2}\\
& = & \Sigma^2(\delta\cdot\partial_H(K_1(K_1\Gamma)\parallel K^2x)) & \text{by 3.2(d)}\\
& = & \Sigma^2\delta & \text{by A7}\\
& = & \delta & \text{by 3.4(iv)}
\end{array}$$

□

Proposition 4.6. Let $\Gamma\equiv\partial_H(((T\cdot K^2y)\parallel((K^2i\cdot s_5+K^2(i\cdot s_\perp))\cdot K^2M))\parallel K^2R)$. Then

$$\begin{array}{rl}
\mathcal{E}_{\mathrm{SABP}}\cup\mathrm{SC}\vdash\Gamma = & K^2i\cdot\partial_H((s_5\cdot K^2M)\parallel((T\cdot K^2y)\parallel K^2R))\\
& +K^2i\cdot\partial_H((K^2s_\perp\cdot K^2M)\parallel((T\cdot K^2y)\parallel K^2R))
\end{array}$$

Proof.

$$\begin{array}{lll}
\Gamma & = & \partial_H(((T\cdot K^2y)\parallel((K^2i\cdot s_5+K^2(i\cdot s_\perp))\cdot K^2M))\mathbin{\lfloor\!\lfloor} K^2R) \\
& & +\,\partial_H(((T\cdot K^2y)\parallel((K^2i\cdot s_5+K^2(i\cdot s_\perp))\cdot K^2M))\mid K^2R) \qquad \text{by CM1, D3, 4.5(vii), A6} \\
& = & \partial_H(((T\cdot K^2y)\parallel(K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M))\mathbin{\lfloor\!\lfloor} K^2R) \\
& & +\,\partial_H(((T\cdot K^2y)\parallel(K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M))\mid K^2R) \qquad \text{by 3.2(a), A4} \\
& = & \partial_H(((T\cdot K^2y)\mathbin{\lfloor\!\lfloor}(K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M))\mathbin{\lfloor\!\lfloor} K^2R) \\
& & +\,\partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y))\mathbin{\lfloor\!\lfloor} K^2R) \\
& & +\,\partial_H(((T\cdot K^2y)\mathbin{\lfloor\!\lfloor}(K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M))\mid K^2R) \\
& & +\,\partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y))\mid K^2R) \qquad \text{by CM1, 4.5(ii), A6, CM4, 8, D3} \\
& = & \partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y))\mathbin{\lfloor\!\lfloor} K^2R) \\
& & +\,\partial_H(((T\cdot K^2y)\mathbin{\lfloor\!\lfloor}(K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M))\mid K^2R) \\
& & +\,\partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y))\mid K^2R) \qquad \text{by SC1, 4.5(iii), A1, 6} \\
& = & \partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y))\mathbin{\lfloor\!\lfloor} K^2R) \\
& & +\,\partial_H(((T\cdot K^2y)\mid K^2R)\mathbin{\lfloor\!\lfloor}(K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)) \\
& & +\,\partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mid K^2R)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y)) \qquad \text{by SC3, 5} \\
& = & \partial_H(((K^2i\cdot s_5\cdot K^2M+K^2i\cdot K^2s_\perp\cdot K^2M)\mathbin{\lfloor\!\lfloor}(T\cdot K^2y))\mathbin{\lfloor\!\lfloor} K^2R) \qquad \text{by 4.5(v), (vi), CM2, A6, 7} \\
& = & K^2i\cdot\partial_H((s_5\cdot K^2M)\parallel((T\cdot K^2y)\parallel K^2R)) \\
& & +\,K^2i\cdot\partial_H((K^2s_\perp\cdot K^2M)\parallel((T\cdot K^2y)\parallel K^2R)) \qquad \text{by SC1, 4.5(iv)}
\end{array}$$

□

The fourth and fifth linearization step are quite alike. They are dealt with in Proposition 4.8.

Lemma 4.7.

(i) For $t\in\{s_5, K^2s_\perp, Ks_2\}$, $\mathcal{E}_{\text{SABP}}\vdash\partial_H((t\cdot x)\mathbin{\lfloor\!\lfloor} y)=\delta$.

(ii) $\mathcal{E}_{\text{SABP}}\vdash(s_5\cdot x)\mid(T\cdot y)=\delta$

(iii) $\mathcal{E}_{\text{SABP}}\vdash(K^2s_\perp\cdot x)\mid K^2R=\delta$

(iv) $\mathcal{E}_{\text{SABP}}\vdash(s_5\cdot x)\mid K^2R=c_5\cdot(x\parallel(BKs_3\cdot Ks_2\cdot K^2R))$

(v) $\mathcal{E}_{\text{SABP}}\vdash(T\cdot x)\mid(K^2s_\perp\cdot y)=K^2c_\perp\cdot((s_4\cdot T\cdot x)\parallel y)$

Proof. We prove (i) for $t\equiv s_5$.

$$\begin{array}{lll}
\partial_H((s_5\cdot x)\mathbin{\lfloor\!\lfloor} y) & = & \partial_H(s_5\cdot(x\parallel y)) \qquad \text{by CM3} \\
& = & \partial_H s_5\cdot\partial_H(x\parallel y) \qquad \text{by D4} \\
& = & \delta\cdot\partial_H(x\parallel y) \qquad \text{by D2} \\
& = & \delta \qquad \text{by A7}
\end{array}$$

(ii)

$$\begin{array}{lll}
(s_5\cdot x)\mid(T\cdot y) & = & (s_5\cdot x)\mid((K^2r_\perp\cdot s_4\cdot T\cdot y)+(Kr_2\cdot y)) \qquad \text{by 4.5(i)} \\
& = & (s_5\cdot x)\mid(K^2r_\perp\cdot s_4\cdot T\cdot y)+(s_5\cdot x)\mid(Kr_2\cdot y) \qquad \text{by CM9} \\
& = & \delta+\delta \qquad \text{by } |_{s_5,r_\perp,2,0}, |_{s_5,r_2,1,0} \\
& = & \delta \qquad \text{by A6}
\end{array}$$

(iii)

$$\begin{array}{lll}
(K^2s_\perp\cdot x)\mid K^2R & = & (K^2s_\perp\cdot x)\mid K^2(\Sigma^2(r_5\cdot BKs_3\cdot Ks_2\cdot K^2R)) \qquad \text{by 4.2(iii)} \\
& = & \delta \qquad \text{by 3.6(ii)}
\end{array}$$

(iv)

$$\begin{array}{lll}
(s_5\cdot x)\mid K^2R & = & (s_5\cdot x)\mid K^2(\Sigma^2(r_5\cdot BKs_3\cdot Ks_2\cdot K^2R)) \qquad \text{by 4.2(iii)} \\
& = & (s_5\mid r_5)\cdot(x\parallel(BKs_3\cdot Ks_2\cdot K^2R)) \qquad \text{by } |_{s_5,r_5,2} \\
& = & c_5\cdot(x\parallel(BKs_3\cdot Ks_2\cdot K^2R)) \qquad \text{by } |_c
\end{array}$$

(v)

$$
\begin{array}{rcll}
(T \cdot y) \mid (K^2 s_\perp \cdot x) & = & ((K^2 r_\perp \cdot s_4 \cdot T \cdot x) + (K r_2 \cdot x)) \mid (K^2 s_\perp \cdot y) & \text{by } 4.5(\text{i}) \\
& = & (K^2 r_\perp \cdot s_4 \cdot T \cdot x) \mid (K^2 s_\perp \cdot y) + (K r_2 \cdot x) \mid (K^2 s_\perp \cdot y) & \text{by CM8} \\
& = & (K^2 r_\perp \mid K^2 s_\perp) \cdot ((s_4 \cdot T \cdot x) \parallel y) + (K r_2 \mid K^2 s_\perp) \cdot (x \parallel y) & \text{by CM7} \\
& = & K^2 (r_\perp \mid s_\perp) \cdot ((s_4 \cdot T \cdot x) \parallel y) & \text{by } 3.2(\text{a}), \\
& & & 3.6(\text{ii}), \text{A6}, 7 \\
& = & K^2 c_\perp \cdot ((s_4 \cdot T \cdot x) \parallel y) & \text{by } \mid_c
\end{array}
$$

□

Proposition 4.8. Let $\Gamma \equiv T \cdot K^2 y$. Then

(i) $\mathcal{E}_{\text{SABP}} \cup \text{SC} \vdash \partial_H((s_5 \cdot K^2 M) \parallel (\Gamma \parallel K^2 R)) = c_5 \cdot \partial_H((K^2 M \parallel (BK s_3 \cdot K s_2 \cdot K^2 R)) \parallel \Gamma)$

(ii) $\mathcal{E}_{\text{SABP}} \cup \text{SC} \vdash \partial_H((K^2 s_\perp \cdot K^2 M) \parallel (\Gamma \parallel K^2 R)) = K^2 c_\perp \cdot \partial_H((s_4 \cdot \Gamma) \parallel K^2 (M \parallel R))$

Proof. (i)

$$
\begin{array}{rcll}
\partial_H((s_5 \cdot K^2 M) \parallel (\Gamma \parallel K^2 R)) & = & \partial_H((\Gamma \parallel K^2 R) \mathbin{|\!\!\lfloor} (s_5 \cdot K^2 M)) & \\
& & + \, \partial_H((s_5 \cdot K^2 M) \mid (\Gamma \parallel K^2 R)) & \text{by CM1}, 4.7(\text{i}), \text{A1}, 6, \text{D3} \\
& = & \partial_H((\Gamma \mathbin{|\!\!\lfloor} K^2 R + K^2 R \mathbin{|\!\!\lfloor} \Gamma) \mathbin{|\!\!\lfloor} (s_5 \cdot K^2 M)) & \\
& & + \, \partial_H((s_5 \cdot K^2 M) \mid (\Gamma \mathbin{|\!\!\lfloor} K^2 R + K^2 R \mathbin{|\!\!\lfloor} \Gamma)) & \text{by CM1}, 4.5(\text{v}), \text{A6} \\
& = & \partial_H((s_5 \cdot K^2 M) \mid (\Gamma \mathbin{|\!\!\lfloor} K^2 R)) & \\
& & + \, \partial_H((s_5 \cdot K^2 M) \mid (K^2 R \mathbin{|\!\!\lfloor} \Gamma)) & \text{by CM4}, 9, \text{SC1}, \text{D3}, \\
& & & 4.5(\text{iii}), (\text{vii}), \text{A1}, 6 \\
& = & \partial_H(((s_5 \cdot K^2 M) \mid \Gamma) \mathbin{|\!\!\lfloor} K^2 R) & \\
& & + \, \partial_H(((s_5 \cdot K^2 M) \mid K^2 R) \mathbin{|\!\!\lfloor} \Gamma) & \text{by SC5} \\
& = & \partial_H(((s_5 \cdot K^2 M) \mid K^2 R) \mathbin{|\!\!\lfloor} \Gamma) & \text{by } 4.7(\text{ii}), \text{CM2}, \text{A1}, 6, 7, \\
& = & \partial_H((c_5 \cdot (K^2 M \parallel (BK s_3 \cdot K s_2 \cdot K^2 R))) \mathbin{|\!\!\lfloor} \Gamma) & \text{by } 4.7(\text{iv}) \\
& = & c_5 \cdot \partial_H((K^2 M \parallel (BK s_3 \cdot K s_2 \cdot K^2 R)) \parallel \Gamma) & \text{by CM3}, \text{D1}, 4
\end{array}
$$

(ii)

$$
\begin{array}{rcll}
\partial_H((K^2 s_\perp \cdot K^2 M) \parallel (\Gamma \parallel K^2 R)) & = & \partial_H((\Gamma \parallel K^2 R) \mathbin{|\!\!\lfloor} (K^2 s_\perp \cdot K^2 M)) & \\
& & + \, \partial_H((K^2 s_\perp \cdot K^2 M) \mid (\Gamma \parallel K^2 R)) & \text{by CM1}, 4.7(\text{i}), \\
& & & \text{A1}, 6, \text{D3} \\
& = & \partial_H((\Gamma \mathbin{|\!\!\lfloor} K^2 R + K^2 R \mathbin{|\!\!\lfloor} \Gamma) \mathbin{|\!\!\lfloor} (K^2 s_\perp \cdot K^2 M)) & \\
& & + \, \partial_H((K^2 s_\perp \cdot K^2 M) \mid (\Gamma \mathbin{|\!\!\lfloor} K^2 R + K^2 R \mathbin{|\!\!\lfloor} \Gamma)) & \text{by CM1}, 4.5(\text{v}), \text{A6} \\
& = & \partial_H((K^2 s_\perp \cdot K^2 M) \mid (\Gamma \mathbin{|\!\!\lfloor} K^2 R)) & \\
& & + \, \partial_H((K^2 s_\perp \cdot K^2 M) \mid (K^2 R \mathbin{|\!\!\lfloor} \Gamma)) & \text{by CM4}, 9, \text{SC1}, \text{D3} \\
& & & 4.5(\text{iii}), (\text{vii}), \text{A6} \\
& = & \partial_H(((K^2 s_\perp \cdot K^2 M) \mid \Gamma) \mathbin{|\!\!\lfloor} K^2 R) & \\
& & + \, \partial_H(((K^2 s_\perp \cdot K^2 M) \mid K^2 R) \mathbin{|\!\!\lfloor} \Gamma) & \text{by SC5} \\
& = & \partial_H((\Gamma \mid (K^2 s_\perp \cdot K^2 M)) \mathbin{|\!\!\lfloor} K^2 R) & \text{by } 4.7(\text{iii}), \text{CM2}, \\
& & & \text{A6}, 7 \\
& = & \partial_H((K^2 c_\perp \cdot ((s_4 \cdot \Gamma) \parallel K^2 M)) \mathbin{|\!\!\lfloor} K^2 R) & \text{by } 4.7(\text{v}) \\
& = & K^2 c_\perp \cdot \partial_H(((s_4 \cdot \Gamma) \parallel K^2 M) \parallel K^2 R) & \text{by CM3}, \text{D4}, \\
& & & 3.2(\text{c}), \text{D1} \\
& = & K^2 c_\perp \cdot \partial_H((s_4 \cdot \Gamma) \parallel (K^2 M \parallel K^2 R)) & \text{by SC6} \\
& = & K^2 c_\perp \cdot \partial_H((s_4 \cdot \Gamma) \parallel K^2 (M \parallel R)) & \text{by } 3.2(\text{a})
\end{array}
$$

□

A last linearization step and we are almost done!

Lemma 4.9.

(i) $\mathcal{E}_{\mathrm{SABP}} \vdash (BKs_3 \cdot x) \mid K^2M = \delta$

(ii) $\mathcal{E}_{\mathrm{SABP}} \vdash (T \cdot x) \mid (BKs_3 \cdot y) = \delta$

(iii) $\mathcal{E}_{\mathrm{SABP}} \vdash (Ks_2 \cdot x) \mid K^2M = \delta$

(iv) $\mathcal{E}_{\mathrm{SABP}} \vdash (Ks_2 \cdot x) \mid (T \cdot y) = Kc_2 \cdot (x \parallel y)$

Proof. (i) and (iii) follow from 4.2(ii) and 3.6(iii) and (ii), respectively.
(ii)

$$\begin{array}{rcll}
(T \cdot x) \mid (BKs_3 \cdot y) & = & ((K^2r_\perp \cdot s_4 \cdot T \cdot x) + (Kr_2 \cdot x)) \mid (BKs_3 \cdot y) & \text{by } 4.5(\text{i}) \\
& = & (K^2r_\perp \cdot s_4 \cdot T \cdot x) \mid (BKs_3 \cdot y) + (Kr_2 \cdot x) \mid (BKs_3 \cdot y) & \text{by CM8} \\
& = & (K^2r_\perp \mid BKs_3) \cdot ((s_4 \cdot T \cdot x) \parallel y) + (Kr_2 \mid BKs_3) \cdot (x \parallel y) & \text{by CM7} \\
& = & \delta & \text{by SC3, T1,} \\
& & & 3.6(\text{iii}), (\text{iv}), \text{A6}, 7
\end{array}$$

(iv)

$$\begin{array}{rcll}
(T \cdot x) \mid (Ks_2 \cdot y) & = & ((K^2r_\perp \cdot s_4 \cdot T \cdot x) + (Kr_2 \cdot x)) \mid (Ks_2 \cdot y) & \text{by } 4.5(\text{i}) \\
& = & (K^2r_\perp \cdot s_4 \cdot T \cdot x) \mid (Ks_2 \cdot y) + (Kr_2 \cdot x) \mid (Ks_2 \cdot y) & \text{by CM8} \\
& = & (K^2r_\perp \mid Ks_2) \cdot ((s_4 \cdot T \cdot x) \parallel y) + (Kr_2 \mid Ks_2) \cdot (x \parallel y) & \text{by CM7} \\
& = & Kc_2 \cdot (x \parallel y) & \text{by } 3.6(\text{ii}), \text{A1}, 6, 7, \\
& & & 3.2(\text{a}), \text{SC3}, \mid_c
\end{array}$$

□

Proposition 4.10.

$$\mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \vdash \partial_H((K^2M \parallel (BKs_3 \cdot Ks_2 \cdot K^2R)) \parallel (T \cdot K^2y)) = BKs_3 \cdot Kc_2 \cdot K^2(\partial_H((y \parallel M) \parallel R))$$

Proof. We put $\Gamma \equiv Ks_2 \cdot K^2R$ and $\Gamma' \equiv T \cdot K^2y$. Then

$$
\begin{array}{lcll}
\partial_H((K^2M \parallel (BKs_3 \cdot \Gamma)) \parallel \Gamma') & = & \partial_H((K^2M \parallel (BKs_3 \cdot \Gamma)) \mathbin{\lfloor\!\lfloor} \Gamma') & \\
 & & \quad + \partial_H((K^2M \parallel (BKs_3 \cdot \Gamma)) \mid \Gamma') & \text{by CM1, D0, 3, 4.5(iii),} \\
 & & & \text{A6} \\
 & = & \partial_H((K^2M \mathbin{\lfloor\!\lfloor} (BKs_3 \cdot \Gamma) & \\
 & & \qquad + (BKs_3 \cdot \Gamma) \mathbin{\lfloor\!\lfloor} K^2M) \mathbin{\lfloor\!\lfloor} \Gamma') & \\
 & & \quad + \partial_H((K^2M \mathbin{\lfloor\!\lfloor} (BKs_3 \cdot \Gamma) & \\
 & & \qquad\quad + (BKs_3 \cdot \Gamma) \mathbin{\lfloor\!\lfloor} K^2M) \mid \Gamma') & \text{by CM1, SC3, 4.9(i), A6} \\
 & = & \partial_H(K^2M \mathbin{\lfloor\!\lfloor} ((BKs_3 \cdot \Gamma) \parallel \Gamma') & \\
 & & \quad + \partial_H((BKs_3 \cdot \Gamma) \mathbin{\lfloor\!\lfloor} (K^2M \parallel \Gamma')) & \\
 & & \quad + \partial_H(\Gamma' \mid K^2M) \mathbin{\lfloor\!\lfloor} (BKs_3 \cdot \Gamma)) & \\
 & & \quad + \partial_H(\Gamma' \mid (BKs_3 \cdot \Gamma)) \mathbin{\lfloor\!\lfloor} K^2M) & \text{by CM4, 8, SC1, 3, 5, D3} \\
 & = & \partial_H((BKs_3 \cdot \Gamma) \mathbin{\lfloor\!\lfloor} (K^2M \parallel \Gamma')) & \text{by 4.5(v), (vii), 4.9(ii),} \\
 & & & \text{CM2, A6, 7, D0} \\
 & = & BKs_3 \cdot \partial_H(\Gamma \parallel (K^2M \parallel \Gamma')) & \text{by CM3, D1, 4, 3.2(c)} \\
 & = & BKs_3 \cdot \partial_H((Ks_2 \cdot K^2R) \parallel (K^2M \parallel \Gamma')) & \\
 & = & BKs_3 \cdot (\partial_H((K^2M \parallel \Gamma') \mathbin{\lfloor\!\lfloor} (Ks_2 \cdot K^2R)) & \\
 & & \qquad\qquad + \partial_H((Ks_2 \cdot K^2R) \mid (K^2M \parallel \Gamma')) & \text{by CM1, D3, 4.7(i), A6} \\
 & = & BKs_3 \cdot (\partial_H(K^2M \mathbin{\lfloor\!\lfloor} (\Gamma' \parallel (Ks_2 \cdot K^2R))) & \\
 & & \qquad\quad + \partial_H(\Gamma' \mathbin{\lfloor\!\lfloor} (K^2M \parallel (Ks_2 \cdot K^2R))) & \\
 & & \qquad\quad + \partial_H(((Ks_2 \cdot K^2R) \mid K^2M) \mathbin{\lfloor\!\lfloor} \Gamma') & \\
 & & \qquad\quad + \partial_H(((Ks_2 \cdot K^2R) \mid \Gamma') \mathbin{\lfloor\!\lfloor} K^2M)) & \text{by CM1, 4, 9, 4.5(v),} \\
 & & & \text{SC1, 3, 5, D3, A6} \\
 & = & BKs_3 \cdot \partial_H((Kc_2 \cdot (K^2y \parallel K^2R)) \mathbin{\lfloor\!\lfloor} K^2M) & \text{by 4.5(iii), (vii), CM2,} \\
 & & & \text{4.9(iii), (iv), A1, 6, 7} \\
 & = & BKs_3 \cdot \partial_H(Kc_2 \cdot ((K^2y \parallel K^2R) \parallel K^2M)) & \text{by CM3} \\
 & = & BKs_3 \cdot Kc_2 \cdot K^2(\partial_H((y \parallel (R \parallel M)) & \text{by D1, 4, 3.2(a), (c), SC6}
\end{array}
$$

□

In the final step of the correctness proof we combine the propositions 4.3, 4.4, 4.6, 4.8 and 4.10, and apply wRSP* and FIR_1.

Theorem 4.11.

$$\mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \cup \mathrm{FIR}_1 \cup \mathrm{wRSP}^* \vdash \tau_I(SABP) = \Sigma(r_1 \cdot s_3)^*\delta$$

Proof. We abbreviate $\partial_H(((T \cdot K^2y) \parallel ((K^2i \cdot s_5 + K^2(i \cdot s_\perp)) \cdot K^2M)) \parallel K^2R)$ by Γ.
First note that

$$
\begin{array}{lcll}
\Gamma & = & K^2i \cdot \partial_H((s_5 \cdot K^2M) \parallel ((T \cdot K^2y) \parallel K^2R)) & \\
 & & \quad + K^2i \cdot \partial_H((K^2s_\perp \cdot K^2M) \parallel ((T \cdot K^2y) \parallel K^2R)) & \text{by 4.6} \\
 & = & K^2i \cdot c_5 \cdot \partial_H((K^2M \parallel (BKs_3 \cdot Ks_2 \cdot K^2R)) \parallel (T \cdot K^2y)) & \\
 & & \quad + K^2i \cdot K^2c_\perp \cdot \partial_H((s_4 \cdot T \cdot K^2y) \parallel K^2(M \parallel R)) & \text{by 4.8} \\
 & = & K^2i \cdot c_5 \cdot BKs_3 \cdot Kc_2 \cdot K^2(\partial_H((y \parallel M) \parallel R)) + K^2i \cdot K^2c_\perp \cdot c_4 \cdot \Gamma & \text{by 4.10, 4.4} \\
 & = & K^2i \cdot K^2c_\perp \cdot c_4 \cdot \Gamma + K^2i \cdot c_5 \cdot BKs_3 \cdot Kc_2 \cdot K^2(\partial_H((y \parallel M) \parallel R)) & \text{by A1}
\end{array}
$$

So by Proposition 4.1 we obtain

(1) $\mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \cup \mathrm{wRSP}^* \vdash \Gamma = (K^2i \cdot K^2c_\perp \cdot c_4)^*(K^2i \cdot c_5 \cdot BKs_3 \cdot Kc_2 \cdot K^2(\partial_H((y \parallel M) \parallel R)))$.

It follows that

$$
\begin{array}{lcll}
\tau_I(\Gamma) & = & \tau_I((K^2i \cdot K^2c_\perp \cdot c_4)^*(K^2i \cdot c_5 \cdot BKs_3 \cdot Kc_2 \cdot K^2(\partial_H((y \parallel M) \parallel R)))) & \text{by (1)} \\
 & = & (K^2\tau \cdot K^2\tau \cdot \tau)^*(K^2\tau \cdot \tau \cdot BKs_3 \cdot K\tau \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R)))) & \text{by SEI5, 3.2(c), TI1, 2, 4} \\
 & = & (\tau \cdot \tau \cdot \tau)^*(\tau \cdot \tau \cdot BKs_3 \cdot \tau \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R)))) & \text{by AE}_\tau \\
 & = & \tau^*(\tau \cdot BKs_3 \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R)))) & \text{by T1} \\
 & = & \tau \cdot BKs_3 \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R))) & \text{by FIR}_1\text{, T1}
\end{array}
$$

Hence

$$(2) \quad \mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \cup \mathrm{FIR}_1 \cup \mathrm{wRSP}^* \vdash \tau_I(\Gamma) = \tau \cdot BKs_3 \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R))).$$

We can now calculate

$$\begin{array}{rcll}
\tau_I(\partial_H(((Gx \cdot y) \parallel M) \parallel R)) & = & \tau_I(\Sigma(r_1 \cdot C(\partial_H((s_4 \cdot T \cdot K^2 y) \parallel K^2(M \parallel R)))x)) & \text{by } 4.3 \\
& = & \tau_I(\Sigma(r_1 \cdot C(c_4 \cdot \Gamma)x)) & \text{by } 4.4 \\
& = & \tau_I(\Sigma(r_1 \cdot Cc_4 x \cdot C\Gamma x)) & \text{by } 3.2(\mathrm{a}), \mathrm{AE}_. \\
& = & \Sigma(r_1 \cdot \tau_I(Cc_4 x) \cdot \tau_I(C\Gamma x)) & \text{by } 3.4(\mathrm{iii}), \mathrm{TI1}, 4 \\
& = & \Sigma(r_1 \cdot \tau_I(Cc_4)x \cdot \tau_I(C\Gamma)x) & \text{by } \mathrm{AE}_\tau \\
& = & \Sigma(r_1 \cdot \tau \cdot C(\tau \cdot BKs_3 \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R))))x) & \text{by } 3.2(\mathrm{c}), (\mathrm{d}), (2), \\
& & & \mathrm{TI2}, \mathrm{AE}_\tau \\
& = & \Sigma(r_1 \cdot C\tau x \cdot C(BKs_3)x \cdot C(K^2(\tau_I(\partial_H((y \parallel M) \parallel R))))x) & \text{by } \mathrm{T1}, 3.2(\mathrm{a}), \mathrm{AE}_. \\
& = & \Sigma(r_1 \cdot \tau \cdot Ks_3 x \cdot K^2(\tau_I(\partial_H((y \parallel M) \parallel R)))x) & \text{by } 3.2(\mathrm{d}), 2.6(\mathrm{b}), \\
& & & 3.6(\mathrm{i}) \\
& = & \Sigma(r_1 \cdot s_3 \cdot K(\tau_I(\partial_H((y \parallel M) \parallel R)))) & \text{by } \mathrm{T1} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \tau_I(\partial_H((y \parallel M) \parallel R)) & \text{by } \Sigma\cdot
\end{array}$$

Therefore

$$(3) \quad \mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \cup \mathrm{FIR}_1 \cup \mathrm{wRSP}^* \vdash \tau_I(\partial_H((Gx \cdot y) \parallel M) \parallel R)) = \Sigma(r_1 \cdot s_3) \cdot \tau_I(\partial_H((y \parallel M) \parallel R)).$$

We have finally arrived at a position where we easily can compute

$$\begin{array}{rcll}
\tau_I(SABP) & = & \tau_I(\partial_H((S \parallel M) \parallel R)) & \text{by SABP} \\
& = & \tau_I(\partial_H((((G0 \cdot G1)^* \delta) \parallel M) \parallel R)) & \text{by S} \\
& = & \tau_I(\partial_H(((G0 \cdot G1 \cdot S) \parallel M) \parallel R)) & \text{by SEI1, S, A6} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \tau_I(\partial_H(((G1 \cdot S) \parallel M) \parallel R)) & \text{by (3)} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3) \cdot \tau_I(\partial_H((S \parallel M) \parallel R)) & \text{by (3)} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3) \cdot \tau_I(SABP) & \text{by SABP} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3) \cdot \tau_I(SABP) + \delta & \text{by A6}
\end{array}$$

So

$$(4) \quad \mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \cup \mathrm{FIR}_1 \cup \mathrm{wRSP}^* \vdash \tau_I(SABP) = (\Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3))^* \delta$$

by a second application of Proposition 4.1. The theorem now follows from the observation that

$$\begin{array}{rcll}
\Sigma(r_1 \cdot s_3)^* \delta & = & \Sigma(r_1 \cdot s_3) \cdot (\Sigma(r_1 \cdot s_3)^* \delta) & \text{by SEI1, A6} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3) \cdot (\Sigma(r_1 \cdot s_3)^* \delta) & \text{by SEI1, A6} \\
& = & \Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3) \cdot (\Sigma(r_1 \cdot s_3)^* \delta) + \delta & \text{by A6}
\end{array}$$

and hence also

$$(5) \quad \mathcal{E}_{\mathrm{SABP}} \cup \mathrm{SC} \cup \mathrm{FIR}_1 \cup \mathrm{wRSP}^* \vdash \Sigma(r_1 \cdot s_3)^* \delta = (\Sigma(r_1 \cdot s_3) \cdot \Sigma(r_1 \cdot s_3))^* \delta$$

by a third application of Proposition 4.1. □

5 Semantical issues

In this section we briefly describe a natural semantics for the system of combinatory process algebra.

We assume a combinatory process specification, $((\mathcal{B}, \mathcal{F}), \mathcal{E})$, to be given and let $\gamma : T(\mathcal{B}, \mathcal{F})_A \times T(\mathcal{B}, \mathcal{F})_A \to T(\mathcal{B}, \mathcal{F})_A$ be a communication function such that

$$\gamma(x,y) = \begin{cases} cz_1 \cdots z_n & \text{if there are } r,s \in \mathcal{F} \text{ with } r \mid s = c \in \mathcal{E} \ \&\ x \equiv rz_1 \cdots z_n \ \&\ y \equiv sz_1 \cdots z_n \\ \delta_A & \text{otherwise.} \end{cases}$$

We start building our model by fixing some family of nonempty sets $(\mathcal{B}_\alpha)_{\alpha \in \mathcal{B}}$ with

(i) $\mathcal{B}_P$ is a model for $\mathrm{ACP}_\tau{}^*(T(\mathcal{B},\mathcal{F})_A, \gamma)$, e.g. a model in rooted τ-bisimulation respectively weak bisimulation semantics (cf.[BK85], [BBP93]), and

(ii) $\mathcal{B}_\alpha \subseteq \mathcal{B}_\beta$ iff $\alpha \subseteq \beta$.

The family $(\mathcal{B}_\alpha)_{\alpha \in \mathcal{B}}$ is intended to interpret the set of basic types $\mathcal{B}$ under preservation of its subtype relation. So, in the minimal case, $(\mathcal{B}_\alpha)_{\alpha \in \mathcal{B}}$ consists solely of the $\mathrm{ACP}_\tau{}^*$-model $\mathcal{B}_P$ with subtypes $\mathcal{B}_A$ and $\mathcal{B}_{A^c}$, and a set $\mathcal{B}_D$. We now consider the so-called *full type structure over* $(\mathcal{B}_\alpha)_{\alpha \in \mathcal{B}}$, that is the family of sets $\mathcal{M} = (\mathcal{D}_\alpha)_{\alpha \in \mathcal{T}(\mathcal{B})}$ where

(i) $\mathcal{D}_\alpha = \mathcal{B}_\alpha$, for $\alpha \in \mathcal{B}$,

(ii) $\mathcal{D}_{\alpha \to \beta} = \mathcal{D}_\beta^{\mathcal{D}_\alpha}$, the collection of all set theoretic functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$.

The combinators I_α, $K_{\alpha,\beta}$ and $S_{\alpha,\beta,\gamma}$ can be interpreted in $\mathcal{M}$ by the functions $F^{\mathcal{M}}_{I,\alpha}$, $F^{\mathcal{M}}_{K,\alpha,\beta}$ and $F^{\mathcal{M}}_{S,\alpha,\beta,\gamma}$ where

$F^{\mathcal{M}}_{I,\alpha}(x) = x$, for all $x \in \mathcal{D}_\alpha$,

$F^{\mathcal{M}}_{K,\alpha,\beta}(x,y) = x$, for all $x \in \mathcal{D}_\alpha, y \in \mathcal{D}_\beta$, and

$F^{\mathcal{M}}_{S,\alpha,\beta,\gamma}(x,y,z) = xz(yz)$, for all $x \in \mathcal{D}_{\alpha \to (\beta \to \gamma)}, y \in \mathcal{D}_{\alpha \to \beta}, z \in \mathcal{D}_\alpha$.

Given this interpretation of the combinators, $\mathcal{M}$ clearly satisfies the axioms of combinatory logic (Table 1), and as $\mathcal{M}$ is extensional itself - that is, a function is uniquely determined by its graph - it also satisfies the axioms of extensionality (Table 2).
All the other operators of combinatory process algebra can be dealt with in the following way: here it suffices to define their interpretation by induction on type formation based on the resources of $\mathcal{B}_P$. We give two examples: the operator $\|_\alpha$ can be interpreted by $F^{\mathcal{M}}_{\|,\alpha}$ where

$F^{\mathcal{M}}_{\|,P}(x,y) = x \parallel y$

with the right-hand $\|$ denoting parallel merge in $\mathcal{B}_P$, the $\mathrm{ACP}_\tau{}^*$-model, and

$F^{\mathcal{M}}_{\|,\alpha \to \beta}(x,y) = \lambda z \in \mathcal{D}_\alpha .\, F^{\mathcal{M}}_{\|,\beta}(xz, yz)$;

we can proceed similarly with $\Sigma_{\alpha,\beta}$ by stipulating

$F^{\mathcal{M}}_{\Sigma,\alpha,P}(x) = \Sigma_{y \in \mathcal{D}_\alpha}(xy)$

where the right-hand expression denotes the arbitrary sum in $\mathcal{B}_P$, and

$F^{\mathcal{M}}_{\Sigma,\alpha,\beta \to \gamma}(x) = \lambda y \in \mathcal{D}_\beta .\, F^{\mathcal{M}}_{\Sigma,\alpha,\gamma}(\lambda z \in \mathcal{D}_\alpha .\, xzy)$.

It should be clear that the axioms of argumentwise evaluation (Table 4) are satisfied under this interpretation. Moreover, as $\mathcal{B}_P$ is a model of the ACP_τ-axioms (Table 5), the SEI-axioms (Table 6), the Σ-axioms (Table 7) and the $|_{\mathcal{F}}$-axioms (Table 8) restricted to type P, it follows by induction on type formation, that the higher-order typed axioms hold in $\mathcal{M}$. So, $\mathcal{M}$ is a model for combinatory process algebra and is a model for $\mathcal{E}$, provided it satisfies the equations contained in $\mathcal{E}$.

6 State combinators

The *state operator* λ, see e.g. [BB88], [BW90], describes processes with an independant global state. It is defined such that $\lambda(s,p)$ represents the execution of process p in state s and can be used to translate computer programs (in a higher order language) into process algebra. In this last section, we shall indicate how this operator can be fitted into the framework of combinatory process algebra.

In our setting, λ is a rather unfortunate notation for a combinator since there is no direct link between its behaviour and λ-abstraction in general. However, as we do not wish to introduce yet another notation, we shall stick to it for the time being. Moreover, we shall not describe this combinator in its generality, but restrict its description to a fragment of the full type structure that is big enough to present the underlying ideas.

We let S be a new type, the type of *states*. The execution of a core atomic action a will effect a specific state, and so we obtain an equation of the form

$$\lambda(s, a \cdot p) = a' \cdot \lambda(s', p).$$

Here, a' is the action that occurs as the result of executing a in state s, and s' is the state that ensues when executing a in state s. This a' and s' depend on a and s, and therefore we need in fact the following three combinators with their accompanying axioms.

	ACT	:	$A^c \to (S \to A)$	the action combinator
	EF	:	$A^c \to (S \to S)$	the effect combinator
	λ	:	$S \to (P \to P)$	the state combinator
(SO1)	$[x]\,\lambda x\Box$	$=$	$\Box$	for $\Box \in \{\delta, \tau\}$
(SO2)	$[x, y^{A^c}]\,\lambda xy$	$=$	$[x, y^{A^c}]\,ACTyx$	
(SO3)	$[x, y]\,\lambda x(\tau \cdot y)$	$=$	$[x, y]\,\tau \cdot \lambda xy$	
(SO4)	$[x, y^{A^c}, z]\,\lambda x(y \cdot z)$	$=$	$[x, y^{A^c}, z]\,ACTyx \cdot \lambda(EFyx)z$	
(SO5)	$[x, y, x]\,\lambda x(y + z)$	$=$	$[x, y, z]\,\lambda xy + \lambda xz$	
(SO6)	$[x, y]\,\lambda x(\Sigma y)$	$=$	$[x, y]\,\Sigma(B(\lambda x)y)$	

Table 14: The signature and axioms for the state combinator.

As an example of the use of the state combinator, we consider the (First in, First out) queue, transmitting incoming data while preserving their order. A specification of such a queue Q^{nm} with input port n and output port m is given in Table 15.

We end this last section with the following instructive question: two queues chained together should behave exactly like one single queue, as long as the internal communications are hidden (cf. e.g. [GV93]). Given the apparatus developed so far, can one prove that

$$Q^{13} = \tau_I(\partial_H(Q^{12} \parallel Q^{23}))$$

where $c_2 = r_2 \mid s_2$, $I = \{c_2\}$ and $H = \{r_2, s_2\}$?

References

[B90] J.C.M. Baeten (ed.). *Applications of Process Algebra.* Cambridge Tracts in Theoretical Computer Science 17. Cambridge University Press, 1990.

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

[BB88] J.C.M. Baeten and J.A. Bergstra. Global renaming operators in concrete process algebra. *Information and Computation*, 78(3):205–245, 1988.

$$
\begin{array}{lrcl}
 & 0,1 & : & \mathbb{B} \\
 & EQ & : & D \to (D \to \mathbb{B}) \\
 & \lhd\ \rhd & : & P \to (\mathbb{B} \to (P \to P)) \\
 & \epsilon & : & S \quad \text{(the initial state)} \\
 & ENQ & : & D \to (S \to S) \\
 & DEQ & : & S \to S \\
 & TOP & : & D \to (S \to \mathbb{B}) \\
 & r_n, s_m & : & D \to A^c \\
 & Q^{nm} & : & P \\
 & & & \\
(\mathrm{EQ0}) & [x]\, EQxx & = & [x]\, 0 \\
(\mathrm{EQ1}) & [x,y]\, EQxy & = & [x,y]\, EQyx \\
(\lhd\ \rhd 0) & [x,y]\, x \lhd 0 \rhd y & = & [x,y]\, x \\
(\lhd\ \rhd 1) & [x,y]\, x \lhd 1 \rhd y & = & [x,y]\, y \\
(\lhd\ \rhd 2) & [x,y]\, x \lhd (EQxy) \rhd y & = & [x,y]\, y \\
(\mathrm{DEQ0}) & DEQ\epsilon & = & \epsilon \\
(\mathrm{DEQ1}) & [x]\, DEQ(ENQx\epsilon) & = & \epsilon \\
(\mathrm{DEQ2}) & [x,y,z]\, DEQ(ENQx(ENQyz)) & = & [x,y,z]\, ENQx(DEQ(ENQyz)) \\
(\mathrm{TOP0}) & [x]\, TOPx\epsilon & = & [x]\, 1 \\
(\mathrm{TOP1}) & [x,y]\, TOPx(ENQy\epsilon) & = & EQ \\
(\mathrm{TOP2}) & [x,y,z,u]\, TOPx(ENQy(ENQzu)) & = & [x,y,z,u]\, TOPx(ENQzu) \\
(\mathrm{ACT}_{r_n}) & [x,y]\, ACT(r_n x)y & = & [x,y]\, r_n x \\
(\mathrm{ACT}_{s_m}) & [x,y]\, ACT(s_m x)y & = & [x,y]\, s_m x \lhd TOPxy \rhd \delta \\
(\mathrm{EF}_{r_n}) & [x]\, EF(r_n x) & = & [x]\, ENQx \\
(\mathrm{EF}_{s_m}) & [x]\, EF(s_m x) & = & DEQ \\
(\mathrm{Q}^{nm}) & Q^{nm} & = & \lambda\epsilon(\Sigma(r_n + s_m)^*\delta)
\end{array}
$$

Table 15: The specification of Q^{nm}.

[BBP93] J. A. Bergstra, I. Bethke and A. Ponse. Process algebra with iteration. Report P9314, Programming Research Group, University of Amsterdam, 1993.

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

[BK85] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37(1):77–121, 1985.

[BK86] J.A. Bergstra and J.W. Klop. Verification of an alternating bit protocol by means of process algebra. In W. Bibel and K.P. Jantke, editors, *Math. Methods of Spec. and Synthesis of Software Systems '85, Math. Research 31*, pages 9–23, Berlin, 1986. Akademie-Verlag.

[BKO87] J.A. Bergstra, J.W. Klop, and E.-R. Olderog. Failures without chaos: a new process semantics for fair abstraction. In M. Wirsing, editor, *Formal Description of Programming Concepts – III, Proceedings of the* 3^{th} *IFIP WG 2.2 working conference,* Ebberup 1986, pages 77–103, Amsterdam, 1987. North-Holland.

[BT84] J.A. Bergstra and J.V. Tucker. Top down design and the algebra of communicating processes. *Science of Computer Programming*, 5(2):171–199, 1984.

[BG93] M. Bezem and J. F. Groote. A formal verification of the Alternating Bit Protocol in the Calculus of Constructions. Logic Group Preprint Series, no. 88, Department of Philosophy, University of Utrecht, 1993.

[BB87] T. Bolognesi and E. Brinksma. Introduction to the ISO Specification Language LOTOS. *Computer Networks and ISDN Systems*,14:25-29. Elsevier Science Publishers, 1987.

[B88] E. Brinksma. *On the design of extended LOTOS - a specification language for open distributed systems.* Ph.D. thesis, University of Twente, 1988.

[CF58] H. B. Curry and R. Feys. *Combinatory Logic. Volume I.* North-Holland, Amsterdam, 1958.

[GP90] J. F. Groote and A. Ponse. The syntax and semantics of μCRL. Technical Report CS-R9076, CWI, Amsterdam, 1990.

[GP91] J. F. Groote and A. Ponse. Proof theory for μCRL. Technical Report DS-P9138, CWI, Amsterdam, 1991.

[GV93] R.J. van Glabbeek and F.W. Vaandrager. Modular specifications in Process Algebra. *Theoretical Computer Science*, 113(2):294-348, 1993.

[HS86] J. R. Hindley and J. P. Seldin. *Introduction to combinators and λ-calculus.* London Mathematical Society Student Texts.1, Cambridge University Press, Cambridge, 1986.

[MV90] S. Mauw and G. J. Veltink. A process specification formalism. *Fundamenta Informaticae*, XIII:85-139, 1990.

[M91] S. Mauw. *A process specification formalism.* Ph.D. thesis, University of Amsterdam, 1991.

[P85] J. Parrow. *Fairness properties in process algebra - with applications in communication protocol verification.* Ph.D. thesis, Dept. of Comp. Sci., Uppsala Univ., 1985.

[S67] L. E. Sanchis. Functionals defined by recursion. *Notre Dame Journal of Formal Logic*, VIII(3):161-174,1967.

[S24] M. Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, 92:305-316, 1924.

[V90] F. W. Vaandrager. Two simple protocols. In J. C. M. Baeten, editor, *Applications of Process Algebra*, pages 23-44, Cambridge Tracts in Theoretical Computer Science 17, Cambridge University Press, 1990.