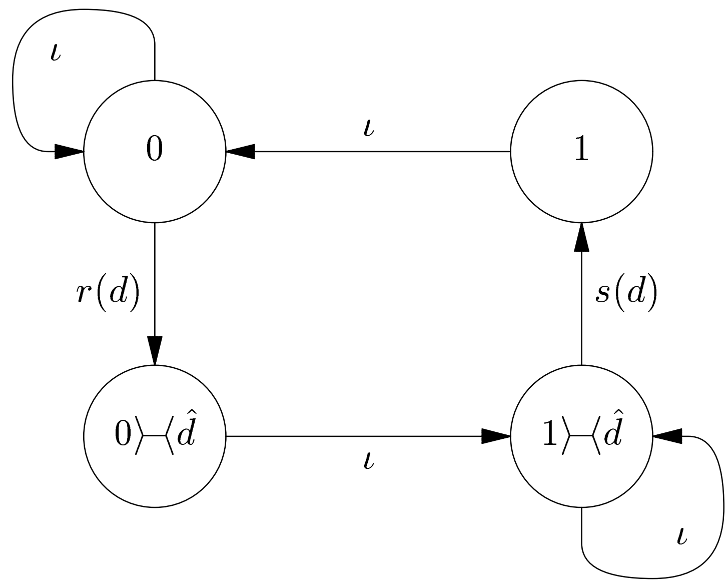
University of Amsterdam
Programming Research Group



Some Verifications in Process Algebra with Iota

Mark van der Zwaag

Report P9806 april 1998

University of Amsterdam
Department of Computer Science
Programming Research Group

Some Verifications in Process Algebra with Iota

Mark van der Zwaag

Report P9806 april 1998

M. B. van der Zwaag

CWI

P.O.Box 94079
1090 GB Amsterdam
The Netherlands

tel. +31 20 592 4203
e-mail: mbz@cwi.nl

Universiteit van Amsterdam, 1998

Contents

Chapter 1

Introduction

Process algebra provides a way to describe the behaviour of systems in an algebraic way. Usually these systems are networks of components that can read, send and communicate data. The application of process algebra lies in the specification of such networks in terms of the specification of their components, and in proving of properties of the thus specified systems.

In this paper, I introduce a special constant ι, called *internal step*, that will be used to represent internal activity. The ideas are from Jan Bergstra and Alban Ponse (cf. [BP98]). The representation of internal activity that is standard in the literature is the *silent step* τ. The silent step originates from [Mil80]. The silent step behaves as a right unit for sequential composition and therefore has a tendency to disappear during verifications. It can be argued that this behaviour is undesirable because we cannot deny the existence of internal activity. The internal step ι does not disappear. Typically, internal activity renamed into ι can be *compressed* to a single ι-step, acting as an indicator of internal activity, but this is not always the case.

I will interpret process expressions in the *frame model*. Although frames have been around for a while, this interpretation has never been made precise.
The introduction of the internal step gives rise to a new notion of equivalence between processes, so-called *rooted orthogonal bisimilarity*. It is called orthogonal, because the internal step is in a way more perpendicular to other actions than the silent step. If you leave the internal step out, usual bisimilarity is obtained.

One of the main reasons to consider this new constant lies in the problems involved in incorporating a priority operator in a process algebra with the silent step (cf. [BBK86, Vaa90, GV92]). We claim that the priority operator θ can be successfully combined with the internal step.

In Chapter 2, I present the signature, axioms and semantics of the algebra. After that I give some preliminaries to protocol verification in Chapter 3. Three well-known simple protocols are then verified in the subsequent chapters. These verifications differ from existing verifications not only in the representation of internal activity but also in the absence of recursive specifications; components are specified with the binary Kleene star operator and during the verifications no recursive specification principles are used. Some pictures illustrating these verifications are included in Appendix A.

Chapter 2

Process Algebra

The Algebra of Communicating Processes (ACP) was introduced in [BK84]. I refer to [BW90] for a treatment. The *binary Kleene star* operator (*) was introduced in the setting of ACP in [BBP94]. In Section 2.2, I present the frame model. I refer to [BP94, BP95] for a treatment of frame algebra. The interpretation of process expressions into the frame model is new. The ideas are from Jan Bergstra and Alban Ponse.

Acknowledgements. I thank Bas Luttik for carefully reading a draft version of this chapter.

2.1 Signature and Axioms

The theory $\mathrm{ACP}^*(A,\gamma)$ consists of the axioms in Table 2.1; it has a finite set A and a partial, commutative and associative function $\gamma : \mathrm{A} \to \mathrm{A} \to \mathrm{A}$ given as parameters. The elements of A are called atomic actions. There is a special constant δ (deadlock). We write A_δ to denote $\mathrm{A} \cup \{\delta\}$. We have binary operators $+$ (alternative composition), $\cdot$ (sequential composition), * (binary Kleene star), $\|$ (parallel composition, merge), $\lfloor\!\lfloor$ (left merge) and $|$ (communication merge). There are unary operators ∂_H (encapsulation), where $H \subseteq \mathrm{A}$. We adopt the following binding conventions: the operator $+$ binds weakest, the operator $\cdot$ binds strongest. We often omit the symbol $\cdot$, writing xy instead of $x \cdot y$. The function γ defines *synchronous* communication between atomic actions, i.e. if two actions communicate, then this communication is an atomic action. If we are talking about communication in a digital system, where usually a communication action is the simultaneous occurrence of a read action and a send action, this comes down to the assumption that transmission takes no time and effort in itself. In most applications we only consider binary, two-party communication, i.e. for all actions $a, b, c \in \mathrm{A}$, $\gamma(\gamma(a,b),c)$ is undefined. This requirement is called *handshaking*.

To obtain the system ACP_ι^* we add the constant ι and the ι-renaming operator ι_I to the signature of ACP^* and the axioms in Table 2.2. We write A_ι for $\mathrm{A} \cup \{\iota\}$ and $\mathrm{A}_{\delta\iota}$ for $\mathrm{A} \cup \{\delta, \iota\}$. The axioms in Table 2.1 remain the same, except that now a and b range over $\mathrm{A}_{\delta\iota}$. The function $\mathrm{init}(x)$ used in the conditional axiom I1 gives the set of initial actions the process associated with x can perform. This function is defined as follows.

$$\mathrm{init}(a) = \{a\} \text{ for all } a \in \mathrm{A}_\iota,\ \mathrm{init}(\delta) = \emptyset,\ \mathrm{init}(xy) = \mathrm{init}(x),$$
$$\mathrm{init}(x+y) = \mathrm{init}(x^*y) = \mathrm{init}(x) \cup \mathrm{init}(y)$$

Table 2.1: The axioms of $\mathrm{ACP}^*(A,\gamma),\ a,b \in \mathrm{A}_\delta,\ H \subseteq \mathrm{A}$

(A1)	$x+y$	$=$	$y+x$	|	(A6)	$x+\delta$	$=$	x		
(A2)	$x+(y+z)$	$=$	$(x+y)+z$	|	(A7)	δx	$=$	δ		
(A3)	$x+x$	$=$	x	|						
(A4)	$(x+y)z$	$=$	$xz+yz$	|	(CF1)	$a \mid b$	$=$	$\gamma(a,b)$	if $\gamma(a,b)\downarrow$	
(A5)	$(xy)z$	$=$	$x(yz)$	|	(CF2)	$a \mid b$	$=$	δ	otherwise	
				|						
(CM1)	$x \parallel y$	$=$	$(x \mathbin{\|\!\lfloor} y + y \mathbin{\|\!\lfloor} x) + x \mid y$	|	(D1)	$\partial_H(a)$	$=$	a	if $a \notin H$	
(CM2)	$a \mathbin{\|\!\lfloor} x$	$=$	ax	|	(D2)	$\partial_H(a)$	$=$	δ	if $a \in H$	
(CM3)	$ax \mathbin{\|\!\lfloor} y$	$=$	$a(x \parallel y)$	|	(D3)	$\partial_H(x+y)$	$=$	$\partial_H(x)+\partial_H(y)$		
(CM4)	$(x+y) \mathbin{\|\!\lfloor} z$	$=$	$x \mathbin{\|\!\lfloor} z + y \mathbin{\|\!\lfloor} z$	|	(D4)	$\partial_H(xy)$	$=$	$\partial_H(x)\cdot\partial_H(y)$		
(CM5)	$ax \mid b$	$=$	$(a \mid b)x$	|	(BKS1)	x^*y	$=$	$x(x^*y)+y$		
(CM6)	$a \mid bx$	$=$	$(a \mid b)x$	|	(BKS2)	$x^*(yz)$	$=$	$(x^*y)z$		
(CM7)	$ax \mid by$	$=$	$(a \mid b)(x \parallel y)$	|						
(CM8)	$(x+y) \mid z$	$=$	$x \mid z + y \mid z$	|	(BKS4)	$\partial_H(x^*y)$	$=$	$\partial_H(x)^*\partial_H(y)$		
(CM9)	$x \mid (y+z)$	$=$	$x \mid y + x \mid z$	|						

Table 2.2: Axioms of $\mathrm{ACP}_\iota^*(A,\gamma)$, $I \subseteq \mathrm{A}$, $a \in \mathrm{A}_{\delta\iota}$

(I1)	$x(\iota(y+z)+z)$	$=$	$x(y+z)$	if	$\mathrm{init}(y)=\{\iota\}$
(IR1)	$\iota_I(a)$	$=$	a	if	$a \notin I$
(IR2)	$\iota_I(a)$	$=$	ι	if	$a \in I$
(IR3)	$\iota_I(x+y)$	$=$	$\iota_I(x)+\iota_I(y)$		
(IR4)	$\iota_I(xy)$	$=$	$\iota_I(x)\cdot\iota_I(y)$		
(BKS5$^\iota$)	$\iota_I(x^*y)$	$=$	$\iota_I(x)^*\iota_I(y)$		

Example 1. An equation derivable using the axiom scheme I1 is $x\iota\iota = x\iota$:

$$\begin{aligned} x\iota\iota &\stackrel{\text{A6}}{=} x(\iota\iota + \delta) \\ &\stackrel{\text{A6}}{=} x(\iota(\iota + \delta) + \delta) \\ &\stackrel{\text{I1}}{=} x(\iota + \delta) \\ &\stackrel{\text{A6}}{=} x\iota \end{aligned}$$

Equations we cannot derive are $x\iota = x$ and $\iota\iota = \iota$. □

Note that the function init is only partially defined, for instance, it is not defined on $\partial_H(p)$. It is defined for so-called basic terms. All terms can be derived equal to a basic term. Therefore, usually functions on processes are defined for basic terms only, and statements on processes are often proved inductively on the structure of basic terms. The subsystem $\{\mathrm{A1} - 7, \mathrm{BKS1}, \mathrm{BKS2}, \mathrm{I1}\}$ is referred to as $\mathrm{BPA}^*_{\delta\iota}$. The operators in its signature are $+, \cdot$ and *.

Definition 2.1. A set of *basic terms* is inductively defined:
i. a is a basic term for all $a \in \mathrm{A}_{\delta\iota}$.
ii. if p, q are basic terms, then for all $a \in \mathrm{A}_\iota$: $a \cdot p$, $p + q$, $p^* q$ are basic terms. □

The following theorem is standard for theories without ι (cf. [BW90]).

Theorem 2.2. ACP^*_ι *is a conservative extension of* $\mathrm{BPA}^*_{\delta\iota}$.

2.1.1 Standard Concurrency

The axioms for *standard concurrency* are in Table 2.3. They are valid for closed terms in the semantics provided below. They can be derived from the other axioms if we adopt the recursive specification principle RSP^* (cf. [BBP94]). Application of these axioms is often left implicit. For instance, we will not write parentheses in expressions like $p \parallel q \parallel r$.

Table 2.3: Standard Concurrency

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

2.1.2 Axioms for Action Alphabets

The alphabet axioms are in Table 2.4. They were introduced in [BBK87]. Contrary to the silent step τ, the internal step ι can be in an alphabet. Note that instances of the alphabet axioms are equations between sets of actions. Therefore they are not axioms in the strict sense, but can be thought of as defining a function. Like the function init, this function is defined for basic terms only.

The conditional alphabet axioms are in Table 2.5. For sets of actions B and C, $B \mid C$ is the set of possible communications between elements of B and elements of C. The axioms with superscript ι are adaptations of their counterparts in ACP_τ.

Table 2.4: Alphabet axioms, $a \in A_\iota$

(AB1)	$\alpha(\delta)$	$=$	$\emptyset$	$\mid$	(AB4)	$\alpha(ax)$	$=$	$\{a\} \cup \alpha(x)$
(AB2)	$\alpha(a)$	$=$	$\{a\}$	$\mid$	(AB5)	$\alpha(x+y)$	$=$	$\alpha(x) \cup \alpha(y)$
				$\mid$	(AB6)	$\alpha(x^* y)$	$=$	$\alpha(x) \cup \alpha(y)$

Table 2.5: Conditional alphabet axioms, $H, I \subseteq A$

(CA1)	$\alpha(x) \mid (\alpha(y) \cap H) \subseteq H$	$\Longrightarrow$	$\partial_H(x \parallel y)$	$=$	$\partial_H(x \parallel \partial_H(y))$
(CA2$^\iota$)	$\alpha(x) \mid (\alpha(y) \cap I) = \emptyset$	$\Longrightarrow$	$\iota_I(x \parallel y)$	$=$	$\iota_I(x \parallel \iota_I(y))$
(CA3)	$\alpha(x) \cap H = \emptyset$	$\Longrightarrow$	$\partial_H(x)$	$=$	x
(CA4$^\iota$)	$\alpha(x) \cap I = \emptyset$	$\Longrightarrow$	$\iota_I(x)$	$=$	x
(CA5)	$H' \subseteq H$	$\Longrightarrow$	$\partial_H(x)$	$=$	$\partial_H \circ \partial_{H'}(x)$
(CA6$^\iota$)	$I' \subseteq I$	$\Longrightarrow$	$\iota_I(x)$	$=$	$\iota_I \circ \iota_{I'}(x)$
(CA7$^\iota$)	$I \cap H = \emptyset$	$\Longrightarrow$	$\iota_I \circ \partial_H(x)$	$=$	$\partial_H \circ \iota_I(x)$

2.2 The Frame Model

An introduction to frames, and to frame algebra, can be found in [BP94, BP95]. I recollect some basic notions in Section 2.2.1.

2.2.1 Frames

We write $\mathbb{F}$ to denote the set of frames and $\mathbb{S}$ to denote the set of states. A frame is a collection of states and labeled transitions between states. The states are generated by an embedding of the naturals, and by the pairing function $.\rangle\!\!-\!\!\langle.$ on $\mathbb{S}^2$. We simply write $0, 1, 2, \ldots$ to represent the embedded naturals. We have an embedding $emb : \mathbb{S} \to \mathbb{F}$. We simply write s instead of $emb(s)$, e.g. $0\rangle\!\!-\!\!\langle 1$ denotes the frame $emb(0\rangle\!\!-\!\!\langle 1)$.
Let $\mathcal{A}$ be a finite set of transition labels. There is a special label ι. We write $\mathcal{A}_\iota$ to denote $\mathcal{A} \cup \{\iota\}$. Let $\gamma : \mathcal{A} \to \mathcal{A} \to \mathcal{A}$ be a partial, associative and commutative function. For all $a \in \mathcal{A}_\iota$ and for all $s, s' \in \mathbb{S}$, the transition $s \xrightarrow{a} s'$ is a frame. Embedded states s and transitions $s \xrightarrow{a} s'$ are atomic frames. The empty frame $\emptyset$ is a constant. We have a binary constructor $\oplus : \mathbb{F} \to \mathbb{F} \to \mathbb{F}$ taking the union of the states and transitions of its arguments. We call $F \oplus G$ the alternative composition of frames F and G.
The set of states of a frame F is denoted $|F|$.

Definition 2.3 (*Operations on* $\mathbb{F}$)**.**

Concurrent Frame Composition

The frame $F \otimes G$ is defined as follows.
$|F \otimes G| = \{s\rangle\!\!-\!\!\langle t \mid s \in |F|,\ t \in |G|\}$,
for all $s \xrightarrow{a} s' \in F$, $t \in |G|$: $s\rangle\!\!-\!\!\langle t \xrightarrow{a} s'\rangle\!\!-\!\!\langle t \in F \otimes G$,
for all $t \xrightarrow{a} t' \in G$, $s \in |F|$: $s\rangle\!\!-\!\!\langle t \xrightarrow{a} s\rangle\!\!-\!\!\langle t' \in F \otimes G$, and
for all $s \xrightarrow{b} s' \in F$, $t \xrightarrow{c} t' \in G$: if $\gamma(b, c) = a$ then $s\rangle\!\!-\!\!\langle t \xrightarrow{a} s'\rangle\!\!-\!\!\langle t' \in F \otimes G$.
There are no other transitions in $F \otimes G$.

Encapsulation
$\partial_H(F)$ is defined by removing all transitions labeled by an element of H.

Abstraction
$\iota_I(F)$ is defined by replacing all transitions with labels from I by transitions labeled with ι.

□

2.2.2 Pointed Frames and Bisimulation

Pointed frames are triples (r, t, F), where r and t are states and F is a frame. We think of r as the root state of the process represented by the pointed frame, and of t as its endpoint. We do not require that r and t are states of F, but r usually is. Consider for example the pointed frame $(0, 1, 0 \xrightarrow{a} 0)$. From its root 0 an a-step to 0 is possible, but it is impossible to terminate. We write $\mathbb{PF}$ to denote the class of pointed frames and $\mathcal{F}, \mathcal{F}', \mathcal{G}, \ldots$ to denote elements of $\mathbb{PF}$.

Definition 2.4. Let $R \subseteq (|F| \cup \{r_f\}) \times (|G| \cup \{r_g\})$.
R is a *bisimulation* between pointed frames (r_f, t_f, F) and (r_g, t_g, G) if it satisfies:

i. $\langle r_f, r_g \rangle \in R$.

ii. If sRu, then $s \xrightarrow{a} t_f$ iff $u \xrightarrow{a} t_g$.

iii. If sRu, $s \xrightarrow{a} s'$ and $s' \neq t_f$, then for some u', $u \xrightarrow{a} u'$ and $s'Ru'$.

iv. If sRu, $u \xrightarrow{a} u'$ and $u' \neq t_g$, then for some s', $s \xrightarrow{a} s'$ and $s'Ru'$.

Pointed frames $\mathcal{F}$ and $\mathcal{G}$ are *bisimilar*, notation $\mathcal{F} \leftrightarroweq \mathcal{G}$, if there is a bisimulation between $\mathcal{F}$ and $\mathcal{G}$.
□

Definition 2.5. Let $R \subseteq (|F| \cup \{r_f\}) \times (|G| \cup \{r_g\})$.
R is an *orthogonal* bisimulation between pointed frames (r_f, t_f, F) and (r_g, t_g, G) if it satisfies the conditions of Definition 2.4, with the restriction that in conditions ii.-iv. $a \neq \iota$, and

i. If sRu, then $(\exists s'.\ s \xrightarrow{\iota} s')$ iff $(\exists u'.\ u \xrightarrow{\iota} u')$.

ii. If sRu and $s \xrightarrow{\iota} t_f$, then
- either $u \xrightarrow{\iota} t_g$, or
- there are states $u_1, \ldots, u_k$ such that $u \xrightarrow{\iota} u_1 \xrightarrow{\iota} \ldots \xrightarrow{\iota} u_k \xrightarrow{\iota} t_g$ and $sRu_1, \ldots, sRu_k$.

iii. If sRu, $s \xrightarrow{\iota} s'$ and $s' \neq t_f$, then
- either $s'Ru$, or
- for some u', $u \xrightarrow{\iota} u'$ and $s'Ru'$, or
- there are states $u', u_1, \ldots, u_k$ such that
  $u \xrightarrow{\iota} u_1 \xrightarrow{\iota} \ldots \xrightarrow{\iota} u_k \xrightarrow{\iota} u'$,
  $sRu_1, \ldots, sRu_k$,
  $s'Ru'$.

iv. If sRu and $u \xrightarrow{\iota} t_g$ then,
- either $s \xrightarrow{\iota} t_f$, or
- there are states $s_1, \ldots, s_k$ such that $s \xrightarrow{\iota} s_1 \xrightarrow{\iota} \ldots \xrightarrow{\iota} s_k \xrightarrow{\iota} t_f$, and $s_1Ru, \ldots, s_kRu$.

v. If sRu, $u \xrightarrow{\iota} u'$ and $u' \neq t_g$, then
- either sRu', or
- for some s', $s \xrightarrow{\iota} s'$ and $s'Ru'$, or
- there are states $s', s_1, \ldots, s_k$,such that
  $s \xrightarrow{\iota} s_1 \xrightarrow{\iota} \ldots \xrightarrow{\iota} s_k \xrightarrow{\iota} s'$,
  $s_1Ru, \ldots, s_kRu$,
  $s'Ru'$.

□

Definition 2.6. A relation R is a *rooted* orthogonal bisimulation between (r_f, t_f, F) and (r_g, t_g, G) if it is an orthogonal bisimulation and satisfies:

i. For all $s \neq t_f$: $r_f \xrightarrow{\iota} s \implies \exists u.\, r_g \xrightarrow{\iota} u \ \& \ sRu.$

ii. For all $u \neq t_g$: $r_g \xrightarrow{\iota} u \implies \exists s.\, r_f \xrightarrow{\iota} s \ \& \ sRu.$

Pointed frames $\mathcal{F}$ and $\mathcal{G}$ are called rooted orthogonal bisimilar, notation $\mathcal{F} \underline{\leftrightarrow}_{ro} \mathcal{G}$, if there is a rooted orthogonal bisimulation between $\mathcal{F}$ and $\mathcal{G}$. □

Theorem 2.7. *Rooted orthogonal bisimilarity is an equivalence relation on* $\mathbb{PF}$.

Proof. Easy. □

Example 2. Consider the following pointed frames.

$$\begin{aligned} \mathcal{F} &= (r, t, r \xrightarrow{a} 0 \oplus 0 \xrightarrow{\iota} t \oplus 0 \xrightarrow{a} t) \\ \mathcal{G} &= (r, t, r \xrightarrow{a} 0 \oplus 0 \xrightarrow{\iota} 1 \oplus 0 \xrightarrow{a} t \oplus 1 \xrightarrow{\iota} t \oplus 1 \xrightarrow{a} t) \end{aligned}$$

The relation $R = \{\langle r, r\rangle, \langle 0, 0\rangle, \langle 0, 1\rangle\}$ is a rooted orthogonal bisimulation between $\mathcal{F}$ and $\mathcal{G}$. These pointed frames are depicted in Figure 2.1. Pairs in R are connected by dotted lines. □

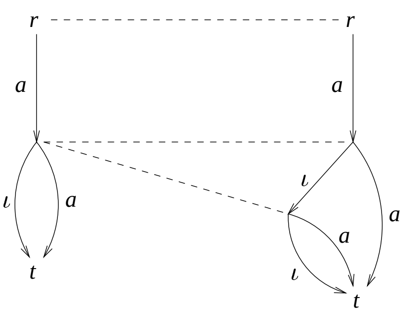


Figure 2.1: Pointed frames $\mathcal{F}$ and $\mathcal{G}$.

Example 3. The first four pointed frames in Figure 2.2 are rooted orthogonal bisimilar. The last two pointed frames are not and neither of them is rooted orthogonal bisimilar to the first four pointed frames. □

2.2.3 Operations and Interpretation

In this section we will define the operators $+, \cdot$, etc, on pointed frames. After that we define an interpretation from ACP_ι^* terms to pointed frames. First we need to define some auxiliary operators. The *state mapping* operator ζ_f renames the states of its argument according to an injective function $f : \mathbb{S} \to \mathbb{S}$. The *root unwinding* operator ρ_r adds a fresh state r to its argument; this state r is the new root and has the same outgoing transitions as the old root, but no incoming transitions. The *endpoint cutting* operator ξ cuts out all transitions leaving from the endpoint of its argument.

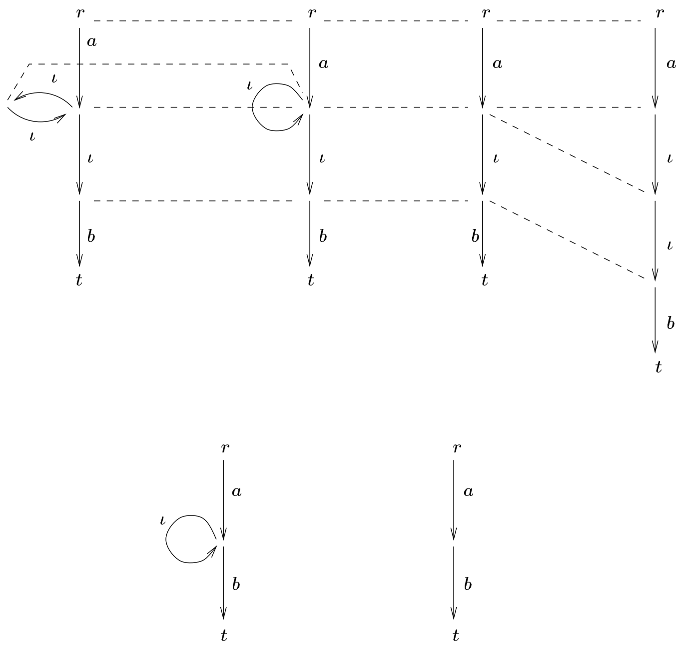


Figure 2.2: Examples

Definition 2.8 (*Auxiliary Operations on* $\mathbb{PF}$)**.**

State Mapping

Let $f : \mathbb{S} \to \mathbb{S}$ be an injective function, then $\zeta_f(s,t,F) = (f(s), f(t), \zeta_f(F))$, where ζ_f on frames is defined by:

$\zeta_f(F_1 \oplus F_2) = \zeta_f(F_1) \oplus \zeta_f(F_2)$,

$\zeta_f(\emptyset) = \emptyset,\ \zeta_f(u) = f(u)$,

$\zeta_f(u \xrightarrow{a} v) = f(u) \xrightarrow{a} f(v)$

Root Unwinding

Let $r \in \mathbb{S}$, then $\rho_r(s,t,F) = (r,t,\rho_r^s(F))$, where the operation ρ_r^s on frames is defined by:

$\rho_r^s(F_1 \oplus F_2) = \rho_r^s(F_1) \oplus \rho_r^s(F_2)$,

$\rho_r^s(\emptyset) = \emptyset,\ \rho_r^s(u) = u$,

$$\rho_r^s(u \xrightarrow{a} v) = \begin{cases} s \xrightarrow{a} v \oplus r \xrightarrow{a} v & \text{if } u = s \\ u \xrightarrow{a} v & \text{otherwise} \end{cases}.$$

Endpoint Cutting

$\xi(s,t,F) = (s,t,\xi_t(F))$, where ξ_t on frames is defined by:

$\xi_t(F_1 \oplus F_2) = \xi_t(F_1) \oplus \xi_t(F_2)$,

$\xi_t(\emptyset) = \emptyset,\ \xi_t(u) = u$,

$$\xi_t(u \xrightarrow{a} v) = \begin{cases} \emptyset & \text{if } u = t \\ u \xrightarrow{a} v & \text{otherwise} \end{cases}.$$

□

For all operations defined in Definition 2.8, it holds that application yields a pointed frame that is rooted orthogonal bisimilar to the argument. This is stated in the following propositions.

Proposition 2.9. *(*State Mapping Invariance*).* $\mathcal{F} \leftrightarroweq_{ro} \zeta_f(\mathcal{F})$.

Proof. If $\mathcal{F} = (r, t, F)$, then $\{\langle u, f(u)\rangle \mid u \in |F| \cup \{r\}\}$ is a rooted orthogonal bisimulation between $\mathcal{F}$ and $\zeta_f(\mathcal{F})$. □

Proposition 2.10. *(*Root Unwinding Invariance*).*
If $\mathcal{F} = (s, t, F)$ *and* $r \notin |F| \cup \{s, t\}$, *then* $\mathcal{F} \leftrightarroweq_{ro} \rho_r(\mathcal{F})$ *and* $\rho_r(\mathcal{F})$ *has no incoming edges at its new root* r.

Proof. State r does not have incoming edges in $\rho^s_r(F)$ because $r \notin |F| \cup \{s, t\}$ and the only transitions involving r in $\rho^s_r(F)$ are leading from r to states in $|F| \cup \{s, t\}$.
The relation $\{\langle u, u\rangle \mid u \in |F|\} \cup \{\langle s, r\rangle\}$ is a rooted orthogonal bisimulation between (s, t, F) and $\rho_r(s, t, F)$. □

Proposition 2.11. *(*Endpoint Cutting Invariance*).* $\mathcal{F} \leftrightarroweq_{ro} \xi(\mathcal{F})$ *and* $\xi(\mathcal{F})$ *has no outgoing edges at its endpoint.*

Proof. Let $\mathcal{F} = (r, t, F)$ and let B be the subset of $|F|$ that can be *reached* from r without passing through t. The relation $\{\langle u, u\rangle \mid u \in B\} \cup \{\langle r, r\rangle\}$ is a rooted orthogonal bisimulation between (r, t, F) and $\xi(r, t, F)$. □

Now we can define the operators $\cdot, +, {}^*, \|, |, \lfloor\!\lfloor, \partial_H$ and ι_I.

Definition 2.12 (*Operators on* $\mathbb{PF}$**).** Let $\mathcal{F} = (s, t, F)$ and $\mathcal{F}' = (s', t', F')$.

1. Let $f : \mathbb{S} \to \mathbb{S}$ be an injective function such that $f(s') = t$ and $|F| \cap f(|F'| \cup \{t'\}) \subseteq \{t\}$, then

$$\mathcal{F} \cdot \mathcal{F}' \quad = \quad (s,\ f(t'),\ \xi_t(F) \oplus \zeta_f(F'))$$

2. Let $r \notin |F| \cup |F'| \cup \{s, t, s', t'\}$ and let $f : \mathbb{S} \to \mathbb{S}$ be an injective function such that $f(r) = r,\ f(t') = t$ and $|F| \cap f(|F'|) \subseteq \{r, t\}$, then

$$\mathcal{F} + \mathcal{F}' \quad = \quad (r,\ t,\ \rho^s_r(F) \oplus \zeta_f \circ \rho^{s'}_r(F'))$$

3. $\mathcal{F} \parallel \mathcal{F}' \ = \ (s \rightarrowtail\!\!\!\!\prec s',\ t \rightarrowtail\!\!\!\!\prec t',\ F \otimes F')$

4. Let $f : \mathbb{S} \to \mathbb{S}$ be an injective function such that $f(t) = t$ and $|F| \cap f(|F'| \cup \{t'\}) \subseteq \{t\}$, then

$$\mathcal{F}^* \mathcal{F}' \quad = \quad (t,\ f(t'),\ \rho^s_t \circ \xi_t(F) \oplus \zeta_f \circ \rho^{s'}_t(F'))$$

5. $\partial_H(\mathcal{F}) = (s, t, \partial_H(F))$

6. Definitions of the left merge, communication merge, and ι-renaming are left to the reader.

□

Theorem 2.13. *$\leftrightarroweq_{ro}$ is a congruence with respect to the operators defined in Definition 2.12.*

Proof. We write $R : \mathcal{F} \leftrightarroweq_{ro} \mathcal{G}$ to express that R is a rooted orthogonal bisimulation between $\mathcal{F}$ and $\mathcal{G}$. Let $\mathcal{F}_1 = (s, t, F_1)$, $\mathcal{F}_2 = (s', t', F_2)$, $\mathcal{G}_1 = (u, v, G_1)$ and $\mathcal{G}_2 = (u', v', G_2)$. Suppose $R_1 : \mathcal{F}_1 \leftrightarroweq_{ro} \mathcal{G}_1$ and $R_2 : \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_2$. We have to show:

1. $\mathcal{F}_1 \cdot \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 \cdot \mathcal{G}_2$
   By definition
   $$\begin{array}{rcll} \mathcal{F}_1 \cdot \mathcal{F}_2 &=& (s,\ f(t'),\ \xi_t(F_1) \oplus \zeta_f(F_2)) & \text{for a suitable function } f \text{, and} \\ \mathcal{G}_1 \cdot \mathcal{G}_2 &=& (u,\ g(v'),\ \xi_v(G_1) \oplus \zeta_g(G_2)) & \text{for a suitable function } g. \end{array}$$
   Now $R_1 \cup \{\langle f(x), g(y)\rangle \mid \langle x, y\rangle \in R_2\} : \mathcal{F}_1 \cdot \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 \cdot \mathcal{G}_2$.
2. $\mathcal{F}_1 + \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 + \mathcal{G}_2$
   By definition
   $$\begin{array}{rcll} \mathcal{F}_1 + \mathcal{F}_2 &=& (r_f,\ t,\ \rho^s_{r_f}(F_1) \oplus \zeta_f \circ \rho^{s'}_{r_f}(F_2)) & \text{for suitable } r_f, f \text{, and} \\ \mathcal{G}_1 + \mathcal{G}_2 &=& (r_g,\ v,\ \rho^u_{r_g}(G_1) \oplus \zeta_g \circ \rho^{u'}_{r_g}(G_2)) & \text{for suitable } r_g, g. \end{array}$$
   Now $\{\langle r, r\rangle\} \cup R_1 \cup \{\langle f(x), g(y)\rangle \mid \langle x, y\rangle \in R_2\} : \mathcal{F}_1 + \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 + \mathcal{G}_2$.
3. $\mathcal{F}_1 \parallel \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 \parallel \mathcal{G}_2$
   By definition
   $$\begin{array}{rcll} \mathcal{F}_1 \parallel \mathcal{F}_2 &=& (s \rangle\!\!\langle s',\ t \rangle\!\!\langle t',\ F_1 \otimes F_2), & \text{and} \\ \mathcal{G}_1 \parallel \mathcal{G}_2 &=& (u \rangle\!\!\langle u',\ v \rangle\!\!\langle v',\ G_1 \otimes G_2). & \end{array}$$
   Now $\{\langle x_f \rangle\!\!\langle y_f,\ x_g \rangle\!\!\langle y_g\rangle \mid \langle x_f, x_g\rangle \in R_1,\ \langle y_f, y_g\rangle \in R_2\} : \mathcal{F}_1 \parallel \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 \parallel \mathcal{G}_2$
4. $\mathcal{F}_1{}^*\mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1{}^*\mathcal{G}_2$
   By definition
   $$\begin{array}{rcll} \mathcal{F}_1{}^*\mathcal{F}_2 &=& (t, f(t'), \rho^s_t \circ \xi_t(F_1) \oplus \zeta_f \circ \rho^{s'}_t(F_2)) & \text{for suitable f, and} \\ \mathcal{G}_1{}^*\mathcal{G}_2 &=& (v, g(v'), \rho^u_v \circ \xi_v(G_1) \oplus \zeta_g \circ \rho^{u'}_v(G_2)) & \text{for suitable g.} \end{array}$$
   Now $R_1 \cup \{\langle t, v\rangle\} \cup \{\langle f(x), g(y)\rangle \mid \langle x, y\rangle \in R_2\} : \mathcal{F}_1{}^*\mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1{}^*\mathcal{G}_2$.
5. $\partial_H(\mathcal{F}_1) \leftrightarroweq_{ro} \partial_H(\mathcal{G}_1)$
   Clearly $R_1 : \partial_H(\mathcal{F}_1) \leftrightarroweq_{ro} \partial_H(\mathcal{G}_1)$.
6. $\mathcal{F}_1 \mathbin{|\!\!\lfloor} \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 \mathbin{|\!\!\lfloor} \mathcal{G}_2$, $\quad \mathcal{F}_1 \mid \mathcal{F}_2 \leftrightarroweq_{ro} \mathcal{G}_1 \mid \mathcal{G}_2$, $\quad \iota_I(\mathcal{F}_1) \leftrightarroweq_{ro} \iota_I(\mathcal{G}_1)$
   Left to the reader.

□

Definition 2.14. Now we define the interpretation $[\![.]\!]$ from $\mathrm{ACP}_\iota^*(\mathrm{A}, \gamma)$ terms to $\mathbb{PF}$. Let $\mathrm{A}_\iota = \mathcal{A}_\iota$ (all atomic actions are labels) and communications defined according to γ.

$$\begin{array}{rcll} [\![a]\!] &=& (0, 1, 0 \xrightarrow{a} 1) & \text{for all } a \in \mathrm{A}_\iota \\ [\![\delta]\!] &=& (0, 1, \emptyset) & \\ [\![x \diamond y]\!] &=& [\![x]\!] \diamond [\![y]\!] & \text{for all } \diamond \in \{\cdot, +, \|, {}^*, \mathbin{|\!\!\lfloor}, |\} \\ [\![\partial_H(x)]\!] &=& \partial_H([\![x]\!]) & \\ [\![\iota_I(x)]\!] &=& \iota_I([\![x]\!]) & \end{array}$$

□

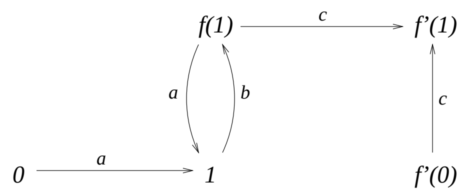
Definition 2.15. We obtain the frame model $\mathbb{PF}/\!\leftrightarrow_{ro}$ by factoring $\leftrightarrow_{ro}$ out on $\mathbb{PF}$. Elements of $\mathbb{PF}/\!\leftrightarrow_{ro}$ are the equivalence classes of $\leftrightarrow_{ro}$ on $\mathbb{PF}$. The equivalence class of a pointed frame $\mathcal{F}$ is denoted $\pi(\mathcal{F})$. □

Theorem 2.16. $\mathbb{PF}/\!\leftrightarrow_{ro} \models \mathrm{ACP}_\iota^*$

Example 4. We give an interpretation of the expression $(ab)^*c$. We use state mappings f and f' and assume that they are suitable. In this case, this means that $f(0) = 1$, $f'(f(1)) = f(1)$, and $f(1), f'(0), f'(1), 0$ and 1 are distinct states.

$$\begin{aligned}
[\![(ab)^*c]\!] &= ((0,1,0 \xrightarrow{a} 1) \cdot (0,1,0 \xrightarrow{b} 1))^*(0,1,0 \xrightarrow{c} 1) \\
&= (0,\ f(1),\ 0 \xrightarrow{a} 1 \oplus 1 \xrightarrow{b} f(1))^*(0,1,0 \xrightarrow{c} 1) \\
&= (f(1),\ f'(1),\ \rho^0_{f(1)} \circ \xi_{f(1)}(0 \xrightarrow{a} 1 \xrightarrow{b} f(1)) \oplus \zeta_{f'} \circ \rho^0_{f(1)}(0 \xrightarrow{c} 1)) \\
&= (f(1),\ f'(1),\ \rho^0_{f(1)}(0 \xrightarrow{a} 1 \xrightarrow{b} f(1)) \oplus \zeta_{f'}(0 \xrightarrow{c} 1 \oplus f(1) \xrightarrow{c} 1)) \\
&= (f(1),\ f'(1),\ f(1) \xrightarrow{a} 1 \oplus 0 \xrightarrow{a} 1 \xrightarrow{b} f(1) \oplus f(1) \xrightarrow{c} f'(1) \oplus f'(0) \xrightarrow{c} f'(1))
\end{aligned}$$

The frame $f(1) \xrightarrow{a} 1 \oplus 0 \xrightarrow{a} 1 \xrightarrow{b} f(1) \oplus f(1) \xrightarrow{c} f'(1) \oplus f'(0) \xrightarrow{c} f'(1)$ is depicted below.



□

2.3 Finite Linear Specifications

In order to distinguish equations that define an abbreviation, we sometimes write $p \equiv q$ to express that p and q are syntactically equal.

Definition 2.17 (*FLS*)**.** A set of derivable equations $\mathcal{L}$ is a *finite linear specification* of a process term p_0 if we can identify process terms $p_1, \ldots, p_n$ such that for each $i \leq n$

$$p_i = \sum_{j \leq n} \alpha_{ij} p_j \ + \beta_i \quad \in \mathcal{L}$$

where all α_{ij}, β_i are non-empty, finite sums of actions from $\mathrm{A}_{\delta\iota}$. We assume that $\mathcal{L}$ contains no other equations.
We write $P(\mathcal{L})$ to denote the set of process terms $\{p_0, \ldots, p_n\}$. □

Note that for an FLS $\mathcal{L}$, the elements of $P(\mathcal{L})$ are not variables, but process terms. Therefore, the equations in $\mathcal{L}$ do not specify a process in the way a recursive linear specification does (cf. [BW90]). Here, the linear format just gives a useful collection of derivable equations between closed process terms, that allows a direct interpretation (see definition below).

Example 5. Consider process terms $p_0 \equiv (ab)^* c$ and $p_1 \equiv b(ab)^* c$.
$\mathcal{L} = \{p_0 = a \cdot p_1 + c,\ p_1 = b \cdot p_0\}$ is a finite linear specification of $(ab)^* c$. □

Example 6. Consider process terms $p_0 \equiv ab \parallel c,\ p_1 \equiv b \parallel c,\ p_2 \equiv ab,\ p_3 \equiv c$ and $p_4 \equiv b$.
Let $\gamma(a, c) = d$, γ undefined otherwise, then

$$\{p_0 = a \cdot p_1 + c \cdot p_2 + d \cdot p_4,\ p_1 = b \cdot p_3 + c \cdot p_4,\ p_2 = a \cdot p_4,\ p_3 = c,\ p_4 = b\}$$

is a finite linear specification of $ab \parallel c$. □

Note that a finite linear specification may contain superfluous equations, namely equations for terms that are not *reachable* from p_0. A *minimal* FLS $\mathcal{L}$ is an FLS such that $P(\mathcal{L})$ does not contain unreachable terms. We assume that specifications are minimal.

Definition 2.18 (*Interpretation of an FLS*)**.** From an FLS $\mathcal{L}$ we construct a frame F as follows.
A function $[.] : P(\mathcal{L}) \to \mathbb{S}$ interprets each $p_i \in P(\mathcal{L})$ as a distinct state.
We take $|F| = \mathit{Range}([.]) \cup \{t\}$ where $t \notin \mathit{Range}([.])$.
For each equation $p_i = \sum \alpha_{ij} p_j\ +\ \beta_i$ in $\mathcal{L}$,
for each summand $b \neq \delta$ in β_i we take $[p_i] \xrightarrow{b} t \in F$, and
for each summand $\alpha_{ij} p_j$; for each summand $a \neq \delta$ in α_{ij} we take $[p_i] \xrightarrow{a} [p_j] \in F$.
There are no other transitions in F.
We write $pf(\mathcal{L})$ to denote the pointed frame $([p_0], t, F)$. □

Theorem 2.19. *For every FLS $\mathcal{L}$ of a process term p_0*

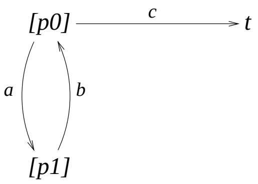
$$pf(\mathcal{L}) \leftrightarroweq_{ro} [\![p_0]\!]$$

Proof. Since all equations in $\mathcal{L}$ are derivable, this follows directly from Theorem 2.16. □

Example 7. Consider the FLS $\mathcal{L}$ from Example 5. We find that

$$pf(\mathcal{L}) = ([p_0], t, [p_0] \xrightarrow{a} [p_1] \oplus [p_0] \xrightarrow{c} t \oplus [p_1] \xrightarrow{b} [p_0])$$

This pointed frame is depicted below. We see that $pf(\mathcal{L}) \leftrightarroweq_{ro} [\![(ab)^* c]\!]$ (cf. Example 4). □



2.4 Fairness Principles

In most applications we distinguish internal actions and external actions. We write I to denote the set of internal actions and abstract from internal activity by placing a process term in the scope of the operator ι_I. We assume that processes behave fairly, i.e. if possible, a process will eventually exit loops of internal actions. In other words, a process with a returning choice $p+q$, where p consists of internal activity only, will eventually choose q. We can formulate *fairness rules* that allow us to derive the desired identities. Here, we define two such rules. They are both sound with respect to rooted orthogonal bisimilarity.

In section 4, we will use the following fairness rule F. Let $I \subseteq A$.The condition $\alpha(y) \subseteq I$ expresses that all activity in y must be internal. It turns out that we need that the exit z necessarily starts with an internal action. It is easy to check that rule F is sound.

$$\text{(F)} \qquad \frac{x = yx + z \quad \alpha(y) \subseteq I \quad init(z) \subseteq I}{\iota \cdot \iota_I(x) = \iota \cdot \iota_I(z)}$$

In section 5 we will encounter internal loops that cannot be expressed in the $x = yx + z$ format. In order to deal with them we will use a variant of the Cluster Fair Abstraction Rule (cf. [BW90]). The CFAR rule is defined for recursive specifications in ACP_τ. Here I define its analog for the setting with linear specifications as defined above in ACP_ι^*. I call this rule CFAR^ι.

Definition 2.20. Let $\mathcal{L}$ be an FLS of p_0. Let $I \subseteq \mathrm{A}$.

We say $C \subseteq P(\mathcal{L})$ is a *cluster of I in $\mathcal{L}$* if $p_0 \in C$ and if for all $p_i \in C$ there exist $i_0, \ldots i_k \in I$; $q_0, \ldots, q_k \in C$; $a_0, \ldots, a_l \in \mathrm{A}_\delta$; $q'_0, \ldots, q'_l \in P(\mathcal{L}) \setminus C$; and $b_0, \ldots, b_m \in \mathrm{A}_\delta$ such that, modulo A1,A2,

$$p_i = \sum_{j \leq k} i_j q_j + \sum_{h \leq l} a_h q'_h + \sum_{o \leq m} b_o \quad \in \mathcal{L}$$

In the above equation the summands $a_h q'_h$ and b_o are the *exits* of p_i. We write $E(p_i)$ to denote the set of exits of p_i. $E = \bigcup_{p \in C} E(p)$ is the set of exits of C.
The cluster is called *conservative* if every exit in E is reachable from every $p \in C$ by doing a number of steps from I.
So, let C be a conservative cluster of I in $\mathcal{L}$, then

$$(\mathrm{CFAR}^\iota) \qquad \frac{\textit{for all } y \in E: \ init(y) \subseteq I}{x \cdot \iota_I(p_0) = x \cdot \iota_I(\sum_{y \in E} y)}$$

□

2.5 Priorities

The priority operator θ was introduced in [BBK86]. Parameter of θ is a partial ordering $<$ on the atomic actions. If an action a is below action b in the ordering, then b has *priority* over a and $\theta(ap+bq)$ can then be derived equal to $b\cdot\theta(q)$. The priority operator can be used to model interrupts in a distributed system. The priority operator is used in the specification of the PAR protocol (see Chapter 6). The axioms are in Table 2.6.

Example 8. Suppose $a < b$.

$$\begin{aligned}
\theta(ap+bq) &= \theta(ap) \lhd bq + \theta(bq) \lhd ap \\
&= \theta(a)\cdot\theta(p) \lhd b + \theta(b)\cdot\theta(q) \lhd a \\
&= a\cdot\theta(p) \lhd b + b\cdot\theta(q) \lhd a \\
&= (a \lhd b)\cdot\theta(p) + (b \lhd a)\cdot\theta(q) \\
&= \delta + b\cdot\theta(q) \\
&= b\cdot\theta(q)
\end{aligned}$$

□

The following example shows that the combination of the priority operator and the silent step τ is inconsistent.

Example 9. In ACP$_\tau$ we have $a\tau a = aa$ by axiom $x\tau = x$ (B1). By the derivations below, we also find that $\theta_{b<a}(a\tau a \parallel b) \neq \theta_{b<a}(aa \parallel b)$. Let $\gamma(a,b)$ be undefined.

$$\begin{aligned}
\theta_{b<a}(a\tau a \parallel b) &= a\cdot\theta_{b<a}(\tau a \parallel b) \\
&= a(\tau ab + b\tau a) \\
\theta_{b<a}(aa \parallel b) &= aab \\
a(\tau ab + b\tau a) &\neq aab
\end{aligned}$$

□

This was one of the reasons to consider the internal step ι. We claim that the axioms in Table 2.6 are consistent with ACP$_\iota^*$.

Table 2.6: Priority axioms, $a, b \in \mathrm{A}_\iota$.

(P1)	$a \lhd b$	$=$	a	$if\ \neg(a < b)$
(P2)	$a \lhd b$	$=$	δ	$if\ a < b$
(P3)	$x \lhd yz$	$=$	$x \lhd y$	
(P4)	$x \lhd (y + z)$	$=$	$(x \lhd y) \lhd z$	
(P5)	$xy \lhd z$	$=$	$(x \lhd z)y$	
(P6)	$(x + y) \lhd z$	$=$	$x \lhd z + y \lhd z$	
(TH1)	$\theta(a)$	$=$	a	
(TH2)	$\theta(xy)$	$=$	$\theta(x) \cdot \theta(y)$	
(TH3)	$\theta(x + y)$	$=$	$\theta(x) \lhd y + \theta(y) \lhd x$	

Chapter 3

Protocol Verification

3.1 Specification

Protocols give descriptions of the behaviour of components in a network. These components are interconnected by ports. If we have expressions for the behaviour of the components, then the behaviour of the network is expressed by the encapsulated merge of these expressions, where the encapsulation enforces communication between the components.
For the description of information flow we need to include data types in our signature. Data are of a different kind than processes. We leave the exact status of data implicit, and simply allow atomic read and send actions to be parameterized with data and with port identifiers. The simultaneous execution of a read and a send action over the same port and the same data element results in a communicate action. This is defined by $\gamma(s_i(d), r_i(d)) = c_i(d)$ for all data elements d and port identifiers i. Usually we take γ undefined otherwise, so the handshaking requirement is satisfied. This kind of communication is referred to as *value-passing*. The parameterization of actions originates from μCRL (cf. [GP94]).

If $D = \{d_0, \ldots, d_n\}$ is a finite set of data elements we write $\sum_{d \in D} p(d)$ to abbreviate the process expression $p[d_0/d] + \ldots + p[d_n/d]$.

Example 1 (*A simple communication protocol*)**.** We consider a network with components S, K, and R, internal ports 2,3 and external ports 1,4, as depicted in Figure 3.1. The channel K is unreliable in the transmission of data elements (from some finite data set D). After receiving a datum from the sender S, the channel has two options: It can send on the datum to the receiver R and proceed accordingly, or it can fail to do anything. We use a dummy internal action i to make this choice non-deterministic. The channel is able to send on acknowledgements reliably. The sender retransmits its message until it receives an acknowledgement.

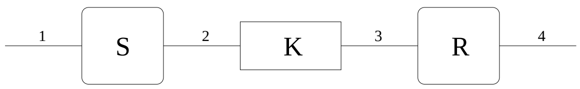


Figure 3.1: A simple architecture

The components are specified as follows.

$$
\begin{array}{lcl}
S & = & (\sum_{d\in D} r_1(d)\cdot(s_2(d)^* r_2(\mathrm{ack})))^*\delta \\
K & = & (\sum_{d\in D} r_2(d)\cdot(i + i\cdot s_3(d)\cdot r_3(\mathrm{ack})\cdot s_2(\mathrm{ack})))^*\delta \\
R & = & (\sum_{d\in D} r_3(d)\cdot s_4(d)\cdot s_3(\mathrm{ack}))^*\delta
\end{array}
$$

The internal actions that are allowed to be executed in communications only, are in

$$H = \{s_m(x), r_m(x) \mid m \in \{2,3\}\ ,\ x \in D \cup \{\mathrm{ack}\}\}$$

and the communication function is defined by $\gamma(s_m(x), r_m(x)) = c_m(x)$ for all $x \in D \cup \{\mathrm{ack}\}$ and port identifiers m, γ is undefined otherwise. Now the behaviour of the system is specified by $\partial_H(S \parallel K \parallel R)$.
□

Our main business in the following sections will be the verification of protocols, i.e. the proving that the described system shows the intended behaviour.

Example 2. We claim the process specified in Example 1 externally behaves as a one-directional buffer with capacity one. Such a buffer, reading data elements from some finite set D from port i, and sending data over port j, is specified by:

$$B^{ij} \quad = \quad (\sum_{d\in D} r_i(d)\cdot s_j(d))^*\delta$$

So we claim the external behaviour of process $\partial_H(S \parallel K \parallel R)$ equals that of process B^{14}. □

3.2 Linearisation

We will in general start a verification by deriving a linear specification for the encapsulated merge describing the system. We expand the merge by applying the expansion theorem defined below, and drive the encapsulation operator inwards. Some summands will be blocked by the encapsulation. They are dropped by application of axiom A6. If we linearise a term by hand we leave a lot of derivation steps, such as the precise application of the expansion theorem and applications of axiom A6, implicit. It should be noted that a complete derivation will take a lot of work. See for example the computer-checked verification of the alternating bit protocol in [BG93]. The alternating bit protocol is a small protocol. Proof techniques for protocol verification will have to improve considerably to handle larger protocols. See for instance [GS95] for work in this field. Linearisations can be done by hand for simple protocols, but it is a tedious job and easy to make mistakes. Therefore, it is desirable to have as much as possible of the linearisation done mechanically.

Definition 3.1 (*Expansion Theorem*)**.** By $[\ \|_{i\in I}\ P_i]$ we abbreviate $(P_{i_1} \parallel \ldots \parallel P_{i_k})$, where $I = \{i_1, \ldots, i_k\}$ is a non-empty set of indices. Then, if $n \geq 3$

$$[\ \|_{i\in\{1,\ldots,n\}}\ P_i] \ =$$

$$\sum_{1\leq j\leq n} P_j \mathbin{⌊\!⌊} [\ \|_{i\in\{1,\ldots,n\}\setminus\{j\}}\ P_i] \ + \sum_{1\leq i<j\leq n} (P_i|P_j) \mathbin{⌊\!⌊} [\ \|_{k\in\{1,\ldots,n\}\setminus\{i,j\}}\ P_k]$$

□

After derivation of an FLS of the encapsulated merge, we can abstract from internal activity and apply fairness on the way.

Example 3. Consider again the process defined in Example 1. Let for all $d \in D$

$$\begin{aligned} p &= \partial_H(S \parallel K \parallel R) \\ p_d &= \partial_H(s_2(d)^* r_2(\text{ack}) \cdot S \parallel (i + i \cdot s_3(d) \cdot r_3(\text{ack}) \cdot s_2(\text{ack})) \cdot K \parallel R) \end{aligned}$$

We derive

$$\begin{aligned} p &= \textstyle\sum_{d \in D} r_1(d) \cdot c_2(d) \cdot p_d \\ p_d &= i \cdot c_2(d) \cdot p_d \ + \ i \cdot c_3(d) \cdot s_4(d) \cdot c_3(\text{ack}) \cdot c_2(\text{ack}) \cdot p \end{aligned}$$

These equations capture an FLS of p. The interpretation of this FLS (see Definition 2.18) is depicted in Figure 3.2.

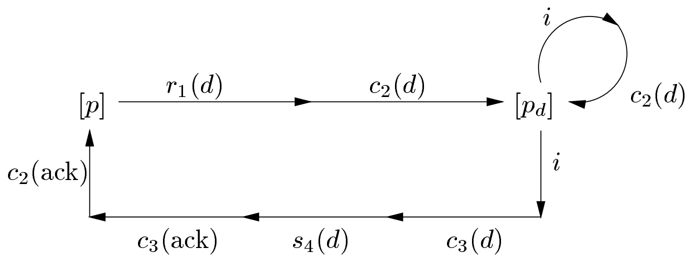


Figure 3.2: The interpretation of p for one datum d

Let I be the set of the internal actions, so $i \in I$ and $c_2(x), c_3(x) \in I$ for all $x \in D \cup \{\text{ack}\}$. We see in Figure 3.2, that the process has a cyclic path of internal actions. We can get rid of it by application of fairness rule F from Section 2.4.

$$\begin{aligned} \iota_I(p) &= \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot \iota_I(p_d) \\ \iota \cdot \iota_I(p_d) &\stackrel{\text{F}}{=} \iota \cdot \iota_I(i \cdot c_3(d) \cdot s_4(d) \cdot c_3(d) \cdot c_2(\text{ack}) \cdot p) \\ &= \iota \cdot \iota \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota \cdot \iota_I(p) \end{aligned}$$

We can eliminate $\iota_I(p_d)$ yielding

$$\iota_I(p) = \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot \iota \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota \cdot \iota_I(p)$$

We can compress internal steps (using the derived identity $x\iota\iota = x\iota$). We get

$$\iota_I(p) = \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota_I(p)$$

□

3.3 Verification

In ACP_τ we can often prove a specification derivable equal to, say, a buffer B^{14}. In ACP_ι this will in general not be possible because traces of internal activity remain visible in expressions that are derived from the specification. We try to reduce our specification to an expression interpretation of which is easily seen to have the intended external behaviour. Such a claim can be substantiated by showing this interpretation *ι-observation bisimilar* to the intended system. This notion is an adaptation of τ-bisimulation (cf. [Mil80, BW90]).

Definition 3.2. Suppose all actions except ι are observable. We write $s \Rightarrow u$ if there is a path of zero or more ι-steps leading from s to u.

The relation $R \subseteq (\{r_f\} \cup |F|) \times (\{r_g\} \cup |G|)$ is a ι-bisimulation between pointed frames (r_f, t_f, F) and (r_g, t_g, G) if R satisfies the following conditions.

i. If $s \xrightarrow{a} s'$, $s' \neq t_f$ and sRu, then either $a = \iota$ and $s'Ru$ or there are states u_1, u_2, u' such that $u \Rightarrow u_1 \xrightarrow{a} u_2 \Rightarrow u'$ and $s'Ru'$.

ii. If $u \xrightarrow{a} u'$, $u' \neq t_g$ and sRu, then either $a = \iota$ and sRu' or there are states s_1, s_2, s' such that $s \Rightarrow s_1 \xrightarrow{a} s_2 \Rightarrow s'$ and $s'Ru'$.

iii. If $s \xrightarrow{a} t_f$ and sRu then either $a = \iota$ and $u \Rightarrow t_g$ or there are states u_1, u_2, u' such that $u \Rightarrow u_1 \xrightarrow{a} u_2 \Rightarrow t_g$.

iv. If $u \xrightarrow{a} t_g$ and sRu then either $a = \iota$ and $s \Rightarrow t_f$ or there are states s_1, s_2, s' such that $s \Rightarrow s_1 \xrightarrow{a} s_2 \Rightarrow t_f$.

Pointed frames $\mathcal{F}$ and $\mathcal{G}$ are called *ι-observation equivalent* or *ι-observation bisimilar* if there is a ι-bisimulation between $\mathcal{F}$ and $\mathcal{G}$. □

Example 4. Consider the equation derived in Example 3. This equation captures an FLS of $\iota_I(p)$. It is clear that the interpretation of this FLS as a pointed frame is ι-observation equivalent to the interpretation of B^{14}. So the described system externally behaves as a buffer as was intended. □

3.4 Redundancy in a Context

In Section 5 some extra machinery is needed. There, we want to reduce the subterms p and q of a term $\partial_H(p \parallel q)$ separately, and after that put the reduced terms back together. Each of these subterms shows behaviour that is redundant in the context $\partial_H(p \parallel q)$; they can perform actions that are blocked by the encapsulation ∂_H. In [Vaa90b, BW90] theory can be found to get rid of such summands using information about the traces of the localization of a subterm in its context. This theory was formulated for ACP_τ, but some minor adjustments will make it work in our case.

Definition 3.3 (*Localization Operator*). Suppose all communications are in the value-passing format.

Let B be a set of atomic actions such that at most one of $s_i(d), r_i(d)$ is in B for every data element d and every communication port i.

Let $C = \{c_i(d) \mid s_i(d) \in B \;\; or \;\; r_i(d) \in B\}$

Let $f : \mathrm{A} \to \mathrm{A}$ be defined by: for all $c_i(d) \in C$:

$$f(c_i(d)) = \begin{cases} s_i(d) & \text{if } s_i(d) \in B; \\ r_i(d) & \text{otherwise;} \end{cases}$$

and for all $a \notin C$: $f(a) = a$.

Then the localization operator ν_B is defined as follows.

$$\nu_B(x) = \rho_f \circ \iota_{\mathrm{A}-(B\cup C)}(x)$$

The axioms for the renaming operator ρ_f are in Table 3.1. $\square$

Table 3.1: Renaming. $f : \mathrm{A} \to \mathrm{A}_{\delta\iota}$

$\rho_f(\iota)$	$=$	ι	|	$\rho_f(x+y)$	$=$	$\rho_f(x)+\rho_f(y)$
$\rho_f(\delta)$	$=$	δ	|	$\rho_f(x\cdot y)$	$=$	$\rho_f(x)\cdot\rho_f(y)$
$\rho_f(a)$	$=$	$f(a)$	|	$\rho_f(x^* y)$	$=$	$\rho_f(x)^*\rho_f(y)$

The localization of p in $\partial_H(p \parallel q)$ is $\nu_{\alpha(p)}(\partial_H(p \parallel q))$. The idea is that we use information about the traces of this localization. Traces are elements of $A_\iota^\star$, the set of all strings of elements of A_ι. We let $\lambda \in \mathrm{A}_\iota^\star$ be the empty string. Equations for the trace set $tr(x)$ of a process x are in Table 3.2.

Table 3.2: Trace sets. $a \in \mathrm{A}_\iota$

$tr(a)$	$=$	$\{\lambda, a\}$	|	$tr(ax)$	$=$	$\{\lambda\} \cup \{a\sigma : \sigma \in tr(x)\}$
$tr(\delta)$	$=$	$\{\lambda\}$	|	$tr(x+y)$	$=$	$tr(x) \cup tr(y)$

Without going into trace set algebra I define, for trace sets Z and actions a:

$$heads(Z) = \{a \in A_\iota \mid a\sigma \in Z\} \quad tails(a, Z) = \{\sigma \in A_\iota^\star \mid a\sigma \in Z\}$$

The axioms for the restriction operator ∇_Z are in Table 3.3.

In [Vaa90b, BW90], the term p occurring in a context $\partial_H(. \parallel q)$ was restricted to a set containing the trace set of its localization as a subset. The situation for $\mathrm{ACP}_\iota{}^*$ is a bit more complicated because the localization operator produces internal steps ι in the localization of p. In ACP_τ there were extra occurrences of τ in the localization, but because τ is not traceable, these occurrences were lost in taking the trace set of the localization. Here, I define a function *tio* taking trace sets to trace sets that throws out all occurrences of ι from the traces in its argument.

Definition 3.4 (*Throw Iotas Out*).

$$tio(Z) = \{\sigma \in A^\star \mid \exists\sigma' \in Z.\ tio(\sigma') = \sigma\}$$

Table 3.3: Restriction. $a \in \mathrm{A}_{\delta\iota}$

$$
\begin{array}{rcl}
\nabla_Z(a) & = & \partial_{\mathrm{A}-heads(Z)}(a) \\
\nabla_Z(ax) & = & \partial_{\mathrm{A}-heads(Z)}(a) \cdot \nabla_{tails(a,Z)}(x) \\
\nabla_Z(x+y) & = & \nabla_Z(x) + \nabla_Z(y)
\end{array}
$$

where $tio : \mathrm{A}_\iota^\star \to \mathrm{A}^\star$ is defined by:

$$
\begin{array}{rcl}
tio(\lambda) & = & \lambda \\
tio(\iota\sigma) & = & tio(\sigma) \\
tio(a\sigma) & = & a \cdot tio(\sigma)
\end{array}
$$

$\square$

Of course, the function tio does not distinguish between occurrences of ι produced by the localization operator and occurrences that were 'already in p'. Therefore we add condition $\iota \notin \alpha(p)$ to the theorem below. This does not affect its applicability where we need it.

Theorem 3.5. *If $\iota \notin \alpha(p)$ and $tio \circ tr \circ \nu_{\alpha(p)}(\partial_H(p \parallel q)) \subseteq Z$ then*

$$\partial_H(p \parallel q) = \partial_H(\nabla_Z(p) \parallel q)$$

Chapter 4

The Alternating Bit Protocol

The Alternating Bit Protocol (ABP) is probably the most widely studied communication protocol in the process algebra literature. It has a surveyable state space, yet it is not so small that verifying it becomes a triviality. The architecture of the ABP is depicted in Figure 4.1. The network consists of a sender, a receiver, and two channels. The channels are unreliable: upon receiving a message, this message may be damaged before the channel has passed it on. If so, the channel is able to send an error message instead. The sender will retransmit its message until it receives an acknowledgement that the information has reached the receiver. The network has two main states; in state 0 the sender sends data accompanied by the bit 0 over port 2, in state 1, data are accompanied by the bit 1. This bit is used by the receiver as an acknowledgement. If the receiver receives an error message or a datum with the wrong bit, it sends this wrong bit as a negative acknowledgement.

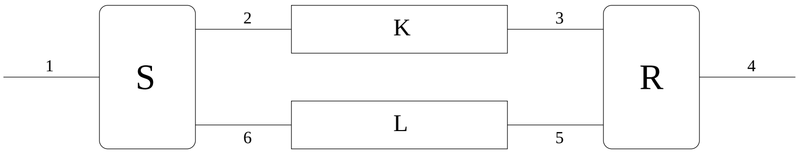


Figure 4.1: Architecture of the Alternating Bit Protocol

4.1 Specification

We assume three disjunct sets D, B, and F, where D is a finite set of data elements, B is the set containing the naturals 0 and 1, and F is the set of frames $D \times B$. For elements n of B, we write $\bar{n}$ to abbreviate $1 - n$.

Description of the components:

The sender S

The sender reads a datum d from port 1, and sends a frame (d, n) to K. It then sends this frame every time it receives the negative acknowledgement $\bar{n}$ or the error message $\perp$ from L until it receives a bit n from L. When the sender has received the bit n, it changes state, reads a new datum e from port 1, sends a frame $(e, \bar{n})$ to K, etc.

Channel K
: Upon receiving a frame from the sender, the channel K can either send this frame to the receiver, or send an error message to the receiver. After that it can read a new frame, etc.

The receiver R
: If the receiver is in state n and reads a frame (d, n) from port 3, it sends d over port 4, sends the bit to L, and changes its state. If it receives an error message or a frame containing the wrong bit $\bar{n}$, it sends this bit $\bar{n}$ to L.

Channel L
: The channel L receives a bit from the receiver. After that it can either send the bit to the sender, or send an error message to the sender. After that it can read a new bit, etc.

These components are specified as follows:

$$
\begin{aligned}
S &= (S_0 \cdot S_1)^* \delta \\
S_n &= \sum_{d\in D} r_1(d) \cdot s_2(dn) \cdot S_{nd} \\
S_{nd} &= ((r_6(\bar{n}) + r_6(\bot)) \cdot s_2(dn))^* r_6(n) \\
K &= (\sum_{x\in F} r_2(x) \cdot (i \cdot s_3(x) + i \cdot s_3(\bot)))^* \delta \\
L &= (\sum_{n\in\{0,1\}} r_5(n) \cdot (i \cdot s_6(n) + i \cdot s_6(\bot)))^* \delta \\
R &= (R_0 \cdot R_1)^* \delta \\
R_n &= ((\sum_{d\in D} r_3(d\bar{n}) + r_3(\bot)) \cdot s_5(\bar{n}))^* \sum_{d\in D} r_3(dn) \cdot s_4(d) \cdot s_5(n)
\end{aligned}
$$

The communication function γ is defined by $\gamma(s_m(x), r_m(x)) = c_m(x)$ for all $m \in \{2, 3, 5, 6\}$ and for all $x \in \{\bot\} \cup F \cup B$. All other communications give δ.

$$
\begin{aligned}
H &= \{s_m(x), r_m(x) \mid m \in \{2, 3, 5, 6\},\ x \in \{\bot\} \cup F \cup B\} \\
I &= \{i\} \cup \{c_m(x) \mid m \in \{2, 3, 5, 6\},\ x \in \{\bot\} \cup F \cup B\} \\
\mathit{ABP} &= \iota_I \circ \partial_H(S \parallel K \parallel R \parallel L)
\end{aligned}
$$

4.2 Requirement

We compare the behaviour of the process ABP to that of a one-element buffer.

$$B^{14} = (\sum_{d\in D} r_1(d) \cdot s_4(d))^* \delta$$

4.3 Linearisation

I use the abbreviations defined below. I write $[p]$ for $\partial_H(p)$.

$$K_{dn} = (i \cdot s_3(dn) + i \cdot s_3(\bot)) \cdot K$$

$$
\begin{aligned}
L_n &= (i \cdot s_6(n) + i \cdot s_6(\perp)) \cdot L \\
P &= \big[S \parallel K \parallel L \parallel R\big] \\
P_d &= \big[S_{0d} \cdot S_1 S \parallel K_{d0} \parallel L \parallel R\big] \\
P'_d &= \big[S_{0d} \cdot S_1 S \parallel K \parallel L_0 \parallel R_1 R\big] \\
Q &= \big[S_1 S \parallel K \parallel L \parallel R_1 R\big] \\
Q_e &= \big[S_{1e} \cdot S \parallel K_{e1} \parallel L \parallel R_1 R\big] \\
Q'_e &= \big[S_{1e} \cdot S \parallel K \parallel L_1 \parallel R\big]
\end{aligned}
$$

The following straightforward calculations yield a linearisation of $\partial_H(S \parallel K \parallel R \parallel L)$.

$$
\begin{aligned}
P &= \textstyle\sum_{d\in D} r_1(d) \cdot \big[s_2(d0) \cdot S_{0d} \cdot S_1 S \parallel K \parallel L \parallel R\big] \\
&= \textstyle\sum_{d\in D} r_1(d) \cdot c_2(d0) \cdot \big[S_{0d} \cdot S_1 S \parallel K_{d0} \parallel L \parallel R\big] \\
&= \textstyle\sum_{d\in D} r_1(d) \cdot c_2(d0) \cdot P_d
\end{aligned}
$$

$$
\begin{aligned}
P_d &= i \cdot \big[S_{0d} \cdot S_1 S \parallel s_3(d0) \cdot K \parallel L \parallel R\big] \; + \\
&\quad\; i \cdot \big[S_{0d} \cdot S_1 S \parallel s_3(\perp) \cdot K \parallel L \parallel R\big] \\
&= i \cdot c_3(d0) \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel L \parallel s_4(d) \cdot s_5(0) \cdot R_1 R\big] \; + \\
&\quad\; i \cdot c_3(\perp) \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel L \parallel s_5(1) \cdot R\big] \\
&= i \cdot c_3(d0) \cdot s_4(d) \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel L \parallel s_5(0) \cdot R_1 R\big] \; + \\
&\quad\; i \cdot c_3(\perp) \cdot c_5(1) \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel L_1 \parallel R\big] \\
&= i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel L_0 \parallel R_1 R\big] \; + \\
&\quad\; i \cdot c_3(\perp) \cdot c_5(1) \cdot (i \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel s_6(1) \cdot L \parallel R\big] \; + \\
&\quad\quad\quad\quad\quad\quad\quad\quad i \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel s_6(\perp) \cdot L \parallel R\big]) \\
&= i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot P'_d \; + \\
&\quad\; i \cdot c_3(\perp) \cdot c_5(1) \cdot (i \cdot c_6(1) \cdot \big[s_2(d0) \cdot S_{0d} \cdot S_1 S \parallel K \parallel L \parallel R\big] \; + \\
&\quad\quad\quad\quad\quad\quad\quad\quad i \cdot c_6(\perp) \cdot \big[s_2(d0) \cdot S_{0d} \cdot S_1 S \parallel K \parallel L \parallel R\big]) \\
&= i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot P'_d \; + \\
&\quad\; i \cdot c_3(\perp) \cdot c_5(1) \cdot (i \cdot c_6(1) \cdot c_2(d0) \cdot \big[S_{0d} \cdot S_1 S \parallel K_{d0} \parallel L \parallel R\big] \; + \\
&\quad\quad\quad\quad\quad\quad\quad\quad i \cdot c_6(\perp) \cdot c_2(d0) \cdot \big[S_{0d} \cdot S_1 S \parallel K_{d0} \parallel L \parallel R)\big]) \\
&= i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot P'_d \; + \\
&\quad\; i \cdot c_3(\perp) \cdot c_5(1) \cdot (i \cdot c_6(1) \cdot c_2(d0) \cdot P_d + i \cdot c_6(\perp) \cdot c_2(d0) \cdot P_d) \\
&= i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot P'_d \; + \\
&\quad\; i \cdot c_3(\perp) \cdot c_5(1) \cdot (i \cdot c_6(1) + i \cdot c_6(\perp)) \cdot c_2(d0) \cdot P_d
\end{aligned}
$$

$$
\begin{aligned}
P'_d &= i \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel s_6(0) \cdot L \parallel R_1 R\big] \; + \\
&\quad\; i \cdot \big[S_{0d} \cdot S_1 S \parallel K \parallel s_6(\perp) \cdot L \parallel R_1 R\big]
\end{aligned}
$$

$$
\begin{aligned}
&= i \cdot c_6(0) \cdot \big[S_1S \parallel K \parallel L \parallel R_1R\big] \;+\\
&\quad\; i \cdot c_6(\perp) \cdot \big[s_2(d0) \cdot S_{0d} \cdot S_1S \parallel K \parallel L \parallel R_1R\big]\\
&= i \cdot c_6(0) \cdot Q \;+\\
&\quad\; i \cdot c_6(\perp) \cdot c_2(d0) \cdot \big[S_{0d} \cdot S_1S \parallel K_{d0} \parallel L \parallel R_1R\big]\\
&= i \cdot c_6(0) \cdot Q \;+\\
&\quad\; i \cdot c_6(\perp) \cdot c_2(d0) \cdot (i \cdot \big[S_{0d} \cdot S_1S \parallel s_3(d0) \cdot K \parallel L \parallel R_1R\big] \;+\\
&\quad\qquad\qquad\qquad\qquad\quad i \cdot \big[S_{0d} \cdot S_1S \parallel s_3(\perp) \cdot K \parallel L \parallel R_1R\big])\\
&= i \cdot c_6(0) \cdot Q \;+\\
&\quad\; i \cdot c_6(\perp) \cdot c_2(d0) \cdot (i \cdot c_3(d0) \cdot \big[S_{0d} \cdot S_1S \parallel K \parallel L \parallel s_5(0) \cdot R_1R\big] \;+\\
&\quad\qquad\qquad\qquad\qquad\quad i \cdot c_3(\perp) \cdot \big[S_{0d} \cdot S_1S \parallel K \parallel L \parallel s_5(0) \cdot R_1R\big])\\
&= i \cdot c_6(0) \cdot Q \;+\\
&\quad\; i \cdot c_6(\perp) \cdot c_2(d0) \cdot (i \cdot c_3(d0) + i \cdot c_3(\perp)) \cdot \big[S_{0d} \cdot S_1S \parallel K \parallel L \parallel s_5(0) \cdot R_1R\big]\\
&= i \cdot c_6(0) \cdot Q \;+\\
&\quad\; i \cdot c_6(\perp) \cdot c_2(d0) \cdot (i \cdot c_3(d0) + i \cdot c_3(\perp)) \cdot c_5(0) \cdot \big[S_{0d} \cdot S_1S \parallel K \parallel L_0 \parallel R_1R\big]\\
&= i \cdot c_6(0) \cdot Q \;+\\
&\quad\; i \cdot c_6(\perp) \cdot c_2(d0) \cdot (i \cdot c_3(d0) + i \cdot c_3(\perp)) \cdot c_5(0) \cdot P'_d
\end{aligned}
$$

We derived the following equations.

$$
\begin{aligned}
P &= \textstyle\sum_{d \in D} r_1(d) \cdot c_2(d0) \cdot P_d\\
P_d &= i \cdot c_3(\perp) \cdot c_5(1) \cdot (i \cdot c_6(1) + i \cdot c_6(\perp)) \cdot c_2(d0) \cdot P_d \;+\\
&\quad\; i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot P'_d\\
P'_d &= i \cdot c_6(\perp) \cdot c_2(d0) \cdot (i \cdot c_3(d0) + i \cdot c_3(\perp)) \cdot c_5(0) \cdot P'_d \;+\; i \cdot c_6(0) \cdot Q
\end{aligned}
$$

In a similar way we can derive

$$
\begin{aligned}
Q &= \textstyle\sum_{e \in D} r_1(e) \cdot c_2(e1) \cdot Q_e\\
Q_e &= i \cdot c_3(\perp) \cdot c_5(0) \cdot (i \cdot c_6(0) + i \cdot c_6(\perp)) \cdot c_2(e1) \cdot Q_e \;+\\
&\quad\; i \cdot c_3(e1) \cdot s_4(e) \cdot c_5(1) \cdot Q'_e\\
Q'_e &= i \cdot c_6(\perp) \cdot c_2(e1) \cdot (i \cdot c_3(e1) + i \cdot c_3(\perp)) \cdot c_5(1) \cdot Q'_e \;+\; i \cdot c_6(1) \cdot P
\end{aligned}
$$

4.4 Abstracting from Internal Activity

We have $ABP \equiv \iota_I(P)$. In this section we abstract from internal activity in P, and apply fairness on the way.

Using the equations from the previous section and the fairness rule F from Section 2.4 we derive:

$$
\iota_I(P) \;=\; \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot \iota_I(P_d)
$$

$$
\begin{aligned}
\iota \cdot \iota_I(P_d) &= \iota \cdot \iota_I(i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(0) \cdot P'_d) \\
&= \iota \cdot \iota \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota_I(P'_d) \\
\iota \cdot \iota_I(P'_d) &= \iota \cdot \iota \cdot \iota \cdot \iota_I(Q)
\end{aligned}
$$

Combining these results we find that

$$\iota_I(P) = \sum_{d \in D} r_1(d) \cdot \iota \cdot \iota \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota \cdot \iota \cdot \iota_I(Q)$$

Compressing the ι-steps we get

$$\iota_I(P) = \sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota_I(Q)$$

In a similar way we find

$$\iota_I(Q) = \sum_{e \in D} r_1(e) \cdot \iota \cdot s_4(e) \cdot \iota \cdot \iota_I(P)$$

We eliminate $\iota_I(Q)$:

$$\iota_I(P) = \sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota \cdot \sum_{e \in D} r_1(e) \cdot \iota \cdot s_4(e) \cdot \iota \cdot \iota_I(P)$$

This equation captures an FLS, interpretation of which as a pointed frame (see Definition 2.18) is rooted orthogonal bisimilar to the interpretation of

$$(\sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota)^* \delta$$

and ι-observation bisimilar to the interpretation of B^{14}. So the behaviour of the process *ABP* is as required.

Chapter 5

The Concurrent Alternating Bit Protocol

The architecture of the Concurrent Alternating Bit Protocol (CABP) is depicted in Figure 5.1. The CABP is a more complicated variant of the ABP.
In contrast to the ABP, the CABP has parallel internal activity, i.e. more than one internal actions can happen at the same time. One can identify two streams of data flow: the transmission of data from the sender to the receiver via channel K, and the transmission of bits, acting as acknowledgements, from the acknowledgement sender to the acknowledgement receiver via channel L. I split the system into its upper half and its lower half. This division stems from [KM90].
Another difference with the ABP is that here the channels can fail to send an error message in case messages are damaged.

The CABP was specified and verified before in [GV89, KM90, Wam95]. I follow the level of abstraction of these specifications.

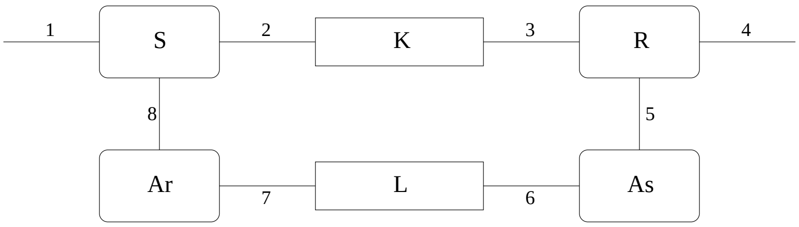


Figure 5.1: The Concurrent Alternating Bit Protocol

5.1 Specification

We assume a finite set D of data elements, a set of bits B containing the naturals 0 and 1, and a set of frames $F = D \times B$. For elements n of B, we write $\bar{n}$ to abbreviate $1 - n$.

Description of the components:

The sender S
: The sender reads a datum d from the input port 1. It thereafter starts sending frames (d, n), where $n \in B$, through port 2. When the sender receives an acknowledgement through port 8, it stops sending frames (d, n), reads a new datum e, and starts sending frames $(e, \bar{n})$ through port 2.

Channel K
: This channel reads frames from port 2. Upon receiving a frame, it can do one of three things: it can pass on the frame to R, it can send an error-message to R, or it can fail to do anything with the received information. In the specification below, i is a dummy internal action used to make the choice between these options non-deterministic.

The receiver R
: If the receiver reads a frame it was waiting for, i.e. a frame containing bit n, it sends the datum through port 4, sends an acknowledgement to As, and starts waiting for a frame containing bit $\bar{n}$. If it reads a wrong frame or an error message, it will remain in the same state.

The acknowledgement sender As
: The acknowledgement sender will repeat sending bits n through port 6, until it reads an acknowledgement from port 5. Then it will start sending bits $\bar{n}$ to L, and so on.

Channel L
: The channel L reads bits from port 6. After that it does (non-deterministically) one of three things: pass on the bit, pass on an error statement, nothing.

The acknowledgement receiver Ar
: The acknowledgement receiver either reads the right bit n, sends an acknowledgement to S, and starts waiting for the other bit, or it reads an error message or the negative acknowledgement $\bar{n}$, and does nothing with it.

These components are specified as follows.

$$
\begin{aligned}
S &= (S_0 \cdot S_1)^* \delta \\
S_n &= \sum_{d \in D} r_1(d) \cdot (s_2(dn)^* r_8(\mathrm{ack})) \\
K &= \left(\sum_{x \in F} r_2(x) \cdot (i \cdot s_3(x) + i \cdot s_3(\perp) + i)\right)^* \delta \\
R &= (R_0 \cdot R_1)^* \delta \\
R_n &= \left(\sum_{d \in D} r_3(d\bar{n}) + r_3(\perp)\right)^* \sum_{d \in D} r_3(dn) \cdot s_4(d) \cdot s_5(\mathrm{ack}) \\
As &= (As_1 \cdot As_0)^* \delta \\
As_n &= s_6(n)^* r_5(\mathrm{ack}) \\
L &= \left(\sum_{n \in B} r_6(n) \cdot (i \cdot s_7(n) + i \cdot s_7(\perp) + i)\right)^* \delta \\
Ar &= (Ar_0 \cdot Ar_1)^* \delta \\
Ar_n &= (r_7(\bar{n}) + r_7(\perp))^* r_7(n) \cdot s_8(\mathrm{ack})
\end{aligned}
$$

We let $\gamma(s_m(x), r_m(x)) = c_m(x)$ for all ports m and all $x \in F \cup B \cup \{\text{ack}, \bot\}$, undefined otherwise.

$$\begin{aligned}
H &= \{s_m(x), r_m(x) \mid m \in \{2,3,5,6,7,8\},\ x \in F \cup B \cup \{\text{ack}, \bot\}\} \\
I &= \{i\} \cup \{c_m(x) \mid m \in \{2,3,5,6,7,8\},\ x \in F \cup B \cup \{\text{ack}, \bot\}\} \\
\mathit{CABP} &= \iota_I \circ \partial_H(S \parallel K \parallel R \parallel As \parallel L \parallel Ar)
\end{aligned}$$

5.2 Requirement

We compare the external behaviour of the process *CABP* with a one-directional, one-element buffer.

5.3 Modularity

Straightforward expansion of the merge in the term *CABP* will yield a very large term – a term too large for us to handle. We therefore split the system into two subsystems: the components S, K and R together form a module we call P, and As, L and Ar form module Q. We can reduce these subsystems before we merge them, and thus get a term of a manageable size.

Let

$$\begin{aligned}
H_P &= \{s_m(x), r_m(x) \mid m \in \{2,3\},\ x \in \{\bot\} \cup F\} \\
I_P &= \{i\} \cup \{c_m(x) \mid m \in \{2,3\},\ x \in \{\bot\} \cup F\} \\
P &= \iota_{I_P} \circ \partial_{H_P}(S \parallel K \parallel R)
\end{aligned}$$

$$\begin{aligned}
H_Q &= \{s_m(x), r_m(x) \mid m \in \{6,7\},\ x \in \{\bot\} \cup B\} \\
I_Q &= \{i\} \cup \{c_m(x) \mid m \in \{6,7\},\ x \in \{\bot\} \cup B\} \\
Q &= \iota_{I_Q} \circ \partial_{H_Q}(As \parallel L \parallel Ar)
\end{aligned}$$

Theorem 5.1. $\mathit{CABP} = \iota_I \circ \partial_H(P \parallel Q)$

Proof.

$$\begin{aligned}
\iota_I\partial_H(S \parallel K \parallel R \parallel As \parallel L \parallel Ar) &\overset{\text{CA5}}{=} \iota_I\partial_H\partial_{H_P}(S \parallel K \parallel R \parallel As \parallel L \parallel Ar) \\
&\overset{\text{CA1}}{=} \iota_I\partial_H(\partial_{H_P}(S \parallel K \parallel R) \parallel As \parallel L \parallel Ar) \\
&\overset{\text{CA5}}{=} \iota_I\partial_H\partial_{H_Q}(\partial_{H_P}(S \parallel K \parallel R) \parallel As \parallel L \parallel Ar) \\
&\overset{\text{CA1}}{=} \iota_I\partial_H(\partial_{H_P}(S \parallel K \parallel R) \parallel \partial_{H_Q}(As \parallel L \parallel Ar)) \\
&\overset{\text{CA7}^{\iota}}{=} \partial_H\iota_I(\partial_{H_P}(S \parallel K \parallel R) \parallel \partial_{H_Q}(As \parallel L \parallel Ar)) \\
&\overset{\text{CA6}^{\iota}}{=} \partial_H\iota_I\iota_{I_P}(\partial_{H_P}(S \parallel K \parallel R) \parallel \partial_{H_Q}(As \parallel L \parallel Ar)) \\
&\overset{\text{CA2}^{\iota}}{=} \partial_H\iota_I\iota_{I_P}(\iota_{I_P}\partial_{H_P}(S \parallel K \parallel R) \parallel \partial_{H_Q}(As \parallel L \parallel Ar)) \\
&\overset{\text{CA6}^{\iota}}{=} \partial_H\iota_I(\iota_{I_P}\partial_{H_P}(S \parallel K \parallel R) \parallel \partial_{H_Q}(As \parallel L \parallel Ar)) \\
&\overset{\text{CA6}^{\iota}}{=} \partial_H\iota_I\iota_{I_Q}(\iota_{I_P}\partial_{H_P}(S \parallel K \parallel R) \parallel \partial_{H_Q}(As \parallel L \parallel Ar)) \\
&\overset{\text{CA2}^{\iota}}{=} \partial_H\iota_I\iota_{I_Q}(\iota_{I_P}\partial_{H_P}(S \parallel K \parallel R) \parallel \iota_{I_Q}\partial_{H_Q}(As \parallel L \parallel Ar))
\end{aligned}$$

$$\begin{aligned} &\overset{\mathrm{CA6}^{\iota}}{=} \quad \partial_H \iota_I (P \parallel Q) \\ &\overset{\mathrm{CA7}^{\iota}}{=} \quad \iota_I \partial_H (P \parallel Q) \end{aligned}$$

□

5.3.1 Redundancy in a Context

It is easy to informally check that in its context, the module P cannot perform an $r_8(\text{ack})$ action before it has performed an $s_5(\text{ack})$ action. This is because module Q cannot communicate an acknowledgement along port 8 before it has received an acknowledgement along port 5. After the module P has performed an $s_5(\text{ack})$ action, it cannot do another $s_5(\text{ack})$ action before it has performed an $r_8(\text{ack})$ action, etc. We use this knowledge of the behaviour of module P in its context to reduce the equations we derive in subsequent sections. We obtain this reduction by application of Theorem 3.5.
In the same way we reduce module Q, using the knowledge that after Q has received an acknowledgement along port 5, it will not receive another acknowledgement along port 5 before it has communicated an acknowledgement along port 8.

5.4 Module P

5.4.1 Abbreviations

Let : $e, d \in D$, $n \in B$, $\bar{n} = 1 - n$,
$K_{dn} = (i \cdot s_3(dn) + i \cdot s_3(\bot) + i) \cdot K$,
$S(0) = S_1 \cdot S$,
$S(1) = S$,
$R(0) = R$,
$R(1) = R_1 \cdot R$.
I use the following abbreviations. Note that $P \equiv \iota_{I_P}(P_0)$.

$$\begin{aligned}
P_n &= \partial_{H_P}(S(\bar{n}) \parallel K \parallel R(n)) \\
P^1_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K \parallel R(n)) \\
P^2_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K_{dn} \parallel R(n)) \\
P^3_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(\bot) \cdot K \parallel R(n)) \\
P^4_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(dn) \cdot K \parallel R(n)) \\
P^5_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
P^6_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
P^7_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K_{dn} \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
P^8_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(dn) \cdot K \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
P^9_{nd} &= \partial_{H_P}(s_2(nd)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(\bot) \cdot K \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
P^{10}_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(dn) \cdot K \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
P^{11}_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(\bot) \cdot K \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
P^{12}_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K_{dn} \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
P^{13}_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K_{dn} \parallel R(\bar{n})) \\
P^{14}_{nd} &= \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel K \parallel R(\bar{n}))
\end{aligned}$$

$$
\begin{array}{rcl}
P^{15}_{nd} & = & \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(dn) \cdot K \parallel R(\bar{n})) \\
P^{16}_{nd} & = & \partial_{H_P}(s_2(dn)^* r_8(\text{ack}) \cdot S(n) \parallel s_3(\perp) \cdot K \parallel R(\bar{n})) \\
P^{17}_{nd} & = & \partial_{H_P}(S(n) \parallel K_{dn} \parallel R(\bar{n})) \\
P^{18}_{nd} & = & \partial_{H_P}(S(n) \parallel s_3(dn) \cdot K \parallel R(\bar{n})) \\
P^{19}_{nd} & = & \partial_{H_P}(S(n) \parallel s_3(\perp) \cdot K \parallel R(\bar{n})) \\
P^{20}_{ned} & = & \partial_{H_P}(s_2(en)^* r_8(\text{ack}) \cdot S(\bar{n}) \parallel K_{dn} \parallel R(\bar{n})) \\
P^{21}_{ned} & = & \partial_{H_P}(s_2(e\bar{n})^* r_8(\text{ack}) \cdot S(\bar{n}) \parallel s_3(dn) \cdot K \parallel R(\bar{n}) \\
P^{22}_{ned} & = & \partial_{H_P}(s_2(e\bar{n})^* r_8(\text{ack}) \cdot S(\bar{n}) \parallel s_3(\perp) \cdot K \parallel R(\bar{n}))
\end{array}
$$

$$
\begin{array}{rcl}
R^1_n & = & \partial_{H_P}(S(n) \parallel K \parallel R(n)) \\
R^2_{nd} & = & \partial_{H_P}(S(n) \parallel K_{dn} \parallel R(n)) \\
R^3_n & = & \partial_{H_P}(S(n) \parallel s_3(\perp) \cdot K \parallel R(n)) \\
R^4_{nd} & = & \partial_{H_P}(S(n) \parallel s_3(dn) \cdot K \parallel R(n)) \\
R^5_{nd} & = & \partial_{H_P}(S(0) \parallel K \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
R^6_n & = & \partial_{H_P}(S(n) \parallel K \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
R^7_{nd} & = & \partial_{H_P}(S(n) \parallel K_{dn} \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
R^8_{nd} & = & \partial_{H_P}(S(n) \parallel s_3(dn) \cdot K \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
R^9_n & = & \partial_{H_P}(S(n) \parallel s_3(\perp) \cdot K \parallel s_4(d) \cdot s_5(\text{ack}) \cdot R(\bar{n})) \\
R^{10}_{nd} & = & \partial_{H_P}(S(n) \parallel s_3(dn) \cdot K \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
R^{11}_n & = & \partial_{H_P}(S(n) \parallel s_3(\perp) \cdot K \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
R^{12}_{nd} & = & \partial_{H_P}(S(n) \parallel K_{dn} \parallel s_5(\text{ack}) \cdot R(\bar{n})) \\
R^{13}_{nd} & = & \partial_{H_P}(S(\bar{n}) \parallel K_{dn} \parallel R(\bar{n})) \\
R^{14}_{nd} & = & \partial_{H_P}(S(\bar{n}) \parallel s_3(dn) \cdot K \parallel R(\bar{n}))
\end{array}
$$

5.4.2 Linearisation

Straightforward calculations yield the following equations. Let f, g be ranging over D, and n ranging over B.

$$
\begin{array}{rcl}
P_n & = & \sum_{f \in D} r_1(f) \cdot P^1_{nf} \\
P^1_{nf} & = & r_8(\text{ack}) \cdot R^1_n + c_2(fn) \cdot P^2_{nf} \\
P^2_{nf} & = & r_8(\text{ack}) \cdot R^2_{nf} + i \cdot P^4_{nf} + i \cdot P^3_{nf} + i \cdot P^1_{nf} \\
P^3_{nf} & = & r_8(\text{ack}) \cdot R^3_n + c_3(\perp) \cdot P^1_{nf} \\
P^4_{nf} & = & r_8(\text{ack}) \cdot R^4_{nf} + c_3(fn) \cdot P^5_{nf} \\
P^5_{nf} & = & r_8(\text{ack}) \cdot R^5_{nf} + c_2(fn) \cdot P^7_{nf} + s_4(f) \cdot P^6_{nf} \\
P^6_{nf} & = & r_8(\text{ack}) \cdot R^6_n + c_2(fn) \cdot P^{12}_{nf} + s_5(\text{ack}) \cdot P^{14}_{nf} \\
P^7_{nf} & = & r_8(\text{ack}) \cdot R^7_{nf} + i \cdot P^8_{nf} + i \cdot P^9_{nf} + i \cdot P^5_{nf} + s_4(f) \cdot P^{12}_{nf} \\
P^8_{nf} & = & r_8(\text{ack}) \cdot R^8_{nf} + s_4(f) \cdot P^{10}_{nf} \\
P^9_{nf} & = & r_8(\text{ack}) \cdot R^9_{nf} + s_4(f) \cdot P^{11}_{nf} \\
P^{10}_{nf} & = & r_8(\text{ack}) \cdot R^{10}_{nf} + s_5(\text{ack}) \cdot P^{15}_{nf}
\end{array}
$$

$$
\begin{array}{rcl}
P_{nf}^{11} & = & r_8(\mathrm{ack}) \cdot R_n^{11} + s_5(\mathrm{ack}) \cdot P_{nf}^{16} \\
P_{nf}^{12} & = & r_8(\mathrm{ack}) \cdot R_{nf}^{12} + i \cdot P_{nf}^{10} + i \cdot P_{nf}^{11} + i \cdot P_{nf}^{6} + s_5(\mathrm{ack}) \cdot P_{nf}^{13} \\
P_{nf}^{13} & = & r_8(\mathrm{ack}) \cdot P_{nf}^{17} + i \cdot P_{nf}^{15} + i \cdot P_{nf}^{16} + i \cdot P_{nf}^{14} \\
P_{nf}^{14} & = & r_8(\mathrm{ack}) \cdot P_{\bar{n}} + c_2(fn) \cdot P_{fn}^{13} \\
P_{nf}^{15} & = & r_8(\mathrm{ack}) \cdot P_{nf}^{18} + c_3(fn) \cdot P_{nf}^{14} \\
P_{nf}^{16} & = & r_8(\mathrm{ack}) \cdot P_{nf}^{19} + c_3(\perp) \cdot P_{nf}^{14} \\
P_{nf}^{17} & = & \sum_{g \in D} r_1(g) \cdot P_{ngf}^{20} + i \cdot P_{nf}^{18} + i \cdot P_{nf}^{19} + i \cdot P_{\bar{n}} \\
P_{nf}^{18} & = & \sum_{g \in D} r_1(g) \cdot P_{ngf}^{21} + c_3(fn) \cdot P_{\bar{n}} \\
P_{nf}^{19} & = & \sum_{g \in D} r_1(g) \cdot P_{ngf}^{22} + c_3(\perp) \cdot P_{\bar{n}} \\
P_{ngf}^{20} & = & r_8(\mathrm{ack}) \cdot R_{nf}^{13} + i \cdot P_{ngf}^{21} + i \cdot P_{ngf}^{22} + i \cdot P_{\bar{n}g}^{1} \\
P_{ngf}^{21} & = & r_8(\mathrm{ack}) \cdot R_{nf}^{14} + c_3(fn) \cdot P_{\bar{n}g}^{1} \\
P_{ngf}^{22} & = & r_8(\mathrm{ack}) \cdot R_1^{3} + c_3(\perp) \cdot P_{\bar{n}g}^{1}
\end{array}
$$

5.4.3 Getting rid of redundant summands

As we have argued in Section 5.3.1, all traces of the localization of P in its context satisfy property S:

$$S(\sigma) : \text{for all } u, v \text{ such that } \sigma = uv \text{ it holds that } N(s_5(\mathrm{ack}), u) \geq N(r_8(\mathrm{ack}), u)$$

Here $N(a, \sigma)$ is the number of the occurrences of a in σ. This statement is based on meta-knowledge. Therefore the verification is not fully algebraic. In an earlier verification [Wam95] the same behaviour was identified and proved redundant.

Now let

$$Z = \{\sigma \in A_\iota{}^* \mid S(\sigma)\}$$

Then $tr \circ \nu_{\alpha(P)} \circ \partial_H(P \parallel Q) \subseteq Z$, and by Theorem 3.5

$$\partial_H(P \parallel Q) = \partial_H(\nabla_Z(P) \parallel Q)$$

Because Z is such that for the restriction of P to Z only actions outside I_P matter, we have

$$\nabla_Z(P) = \nabla_Z \circ \iota_{I_P}(P_0) = \iota_{I_P} \circ \nabla_Z(P_0)$$

For all abbreviations X on the left-hand sides of the equations in section 5.4.2, we write $\tilde{X}$ to abbreviate $\nabla_Z(X)$. The reader is invited to check that the FLS below can be derived using the axioms for restriction.

$$
\begin{array}{rcl}
\tilde{P}_n & = & \sum_{f \in D} r_1(f) \cdot \tilde{P}_{nf}^{1} \\
\tilde{P}_{nf}^{1} & = & c_2(fn) \cdot \tilde{P}_{nf}^{2}
\end{array}
$$

$$
\begin{array}{rcl}
\tilde{P}^2_{nf} & = & i \cdot \tilde{P}^4_{nf} + i \cdot \tilde{P}^3_{nf} + i \cdot \tilde{P}^1_{nf} \\
\tilde{P}^3_{nf} & = & c_3(\perp) \cdot \tilde{P}^1_{nf} \\
\tilde{P}^4_{nf} & = & c_3(fn) \cdot \tilde{P}^5_{nf} \\
\tilde{P}^5_{nf} & = & c_2(fn) \cdot \tilde{P}^7_{nf} + s_4(f) \cdot \tilde{P}^6_{nf} \\
\tilde{P}^6_{nf} & = & c_2(fn) \cdot \tilde{P}^{12}_{nf} + s_5(\mathrm{ack}) \cdot \tilde{P}^{14}_{nf} \\
\tilde{P}^7_{nf} & = & i \cdot \tilde{P}^8_{nf} + i \cdot \tilde{P}^9_{nf} + i \cdot \tilde{P}^5_{nf} + s_4(f) \cdot \tilde{P}^{12}_{nf} \\
\tilde{P}^8_{nf} & = & s_4(f) \cdot \tilde{P}^{10}_{nf} \\
\tilde{P}^9_{nf} & = & s_4(f) \cdot \tilde{P}^{11}_{nf} \\
\tilde{P}^{10}_{nf} & = & s_5(\mathrm{ack}) \cdot \tilde{P}^{15}_{nf} \\
\tilde{P}^{11}_{nf} & = & s_5(\mathrm{ack}) \cdot \tilde{P}^{16}_{nf} \\
\tilde{P}^{12}_{nf} & = & i \cdot \tilde{P}^{10}_{nf} + i \cdot \tilde{P}^{11}_{nf} + i \cdot \tilde{P}^6_{nf} + s_5(\mathrm{ack}) \cdot \tilde{P}^{13}_{nf} \\
\tilde{P}^{13}_{nf} & = & r_8(\mathrm{ack}) \cdot \tilde{P}^{17}_{nf} + i \cdot \tilde{P}^{15}_{nf} + i \cdot \tilde{P}^{16}_{nf} + i \cdot \tilde{P}^{14}_{nf} \\
\tilde{P}^{14}_{nf} & = & r_8(\mathrm{ack}) \cdot \tilde{P}_{\bar{n}} + c_2(fn) \cdot \tilde{P}^{13}_{fn} \\
\tilde{P}^{15}_{nf} & = & r_8(\mathrm{ack}) \cdot \tilde{P}^{18}_{nf} + c_3(fn) \cdot \tilde{P}^{14}_{nf} \\
\tilde{P}^{16}_{nf} & = & r_8(\mathrm{ack}) \cdot \tilde{P}^{19}_{nf} + c_3(\perp) \cdot \tilde{P}^{14}_{nf} \\
\tilde{P}^{17}_{nf} & = & \sum_{g \in D} r_1(g) \cdot \tilde{P}^{20}_{ngf} + i \cdot \tilde{P}^{18}_{nf} + i \cdot \tilde{P}^{19}_{nf} + i \cdot \tilde{P}_{\bar{n}} \\
\tilde{P}^{18}_{nf} & = & \sum_{g \in D} r_1(g) \cdot \tilde{P}^{21}_{ngf} + c_3(fn) \cdot \tilde{P}_{\bar{n}} \\
\tilde{P}^{19}_{nf} & = & \sum_{g \in D} r_1(g) \cdot \tilde{P}^{22}_{ngf} + c_3(\perp) \cdot \tilde{P}_{\bar{n}} \\
\tilde{P}^{20}_{ngf} & = & i \cdot \tilde{P}^{21}_{ngf} + i \cdot \tilde{P}^{22}_{ngf} + i \cdot \tilde{P}^1_{\bar{n}g} \\
\tilde{P}^{21}_{ngf} & = & c_3(fn) \cdot \tilde{P}^1_{\bar{n}g} \\
\tilde{P}^{22}_{ngf} & = & c_3(\perp) \cdot \tilde{P}^1_{\bar{n}g}
\end{array}
$$

5.4.4 Abstraction from Internal Activity

In the previous section we found that $P = \iota_{I_P}(\tilde{P}_0)$. In this section we reduce $\iota_{I_P}(\tilde{P}_0)$. In the scope of ι_{I_P}, $\tilde{P}^1_{nf}$, $\tilde{P}^2_{nf}$ and $\tilde{P}^3_{nf}$ form a cluster with exit $i \cdot \tilde{P}^4_{nf}$.

$$
\begin{array}{rcl}
\iota_{I_P}(\tilde{P}_n) & = & \sum_{f \in D} r_1(f) \cdot \iota_{I_P}(\tilde{P}^1_{nf}) \\
& \stackrel{\mathrm{CFAR}^\iota}{=} & \sum_{f \in D} r_1(f) \cdot \iota_{I_P}(i \cdot \tilde{P}^4_{nf}) \\
& = & \sum_{f \in D} r_1(f) \cdot \iota \cdot \iota_{I_P}(c_3(fn) \cdot \tilde{P}^5_{nf}) \\
& = & \sum_{f \in D} r_1(f) \cdot \iota \cdot \iota \cdot \iota_{I_P}(\tilde{P}^5_{nf}) \\
& = & \sum_{f \in D} r_1(f) \cdot \iota \cdot \iota_{I_P}(\tilde{P}^5_{nf})
\end{array}
$$

Some other equations that can be derived are the following, where 5.1 follows from 5.2, etc.

$$\iota_{I_P}(\tilde{P}^{21}_{ngf}) = \iota_{I_P}(\tilde{P}^{22}_{ngf}) \tag{5.1}$$

$$\iota_{I_P}(\tilde{P}^{18}_{nf}) = \iota_{I_P}(\tilde{P}^{19}_{nf}) \tag{5.2}$$
$$\iota_{I_P}(\tilde{P}^{15}_{nf}) = \iota_{I_P}(\tilde{P}^{16}_{nf}) \tag{5.3}$$
$$\iota_{I_P}(\tilde{P}^{10}_{nf}) = \iota_{I_P}(\tilde{P}^{11}_{nf}) \tag{5.4}$$
$$\iota_{I_P}(\tilde{P}^{8}_{nf}) = \iota_{I_P}(\tilde{P}^{9}_{nf}) \tag{5.5}$$

From 5.1 it follows that

$$\begin{aligned}
\iota_{I_P}(\tilde{P}^{20}_{ngf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{21}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{22}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{1}_{\bar{n}g}) \\
&= \iota \cdot \iota_{I_P}(\tilde{P}^{21}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{1}_{\bar{n}g})
\end{aligned}$$

In a similar way we derive using 5.1–5.5

$$\begin{aligned}
\iota_{I_P}(\tilde{P}^{17}_{nf}) &= \sum\nolimits_{g \in D} r_1(g) \cdot \iota_{I_P}(\tilde{P}^{20}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{18}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}_{\bar{n}}) \\
\iota_{I_P}(\tilde{P}^{13}_{nf}) &= r_8(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{17}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{15}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{14}_{nf}) \\
\iota_{I_P}(\tilde{P}^{12}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{10}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{6}_{nf}) + s_5(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{13}_{nf}) \\
\iota_{I_P}(\tilde{P}^{7}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{8}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{5}_{nf}) + s_4(f) \cdot \iota_{I_P}(\tilde{P}^{12}_{nf})
\end{aligned}$$

Combining these results we get

$$\begin{aligned}
\iota_{I_P}(\tilde{P}_{n}) &= \sum\nolimits_{f \in D} r_1(f) \cdot \iota_{I_P}(\tilde{P}^{1}_{nf}) \\
\iota_{I_P}(\tilde{P}^{1}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{5}_{nf}) \\
\iota_{I_P}(\tilde{P}^{5}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{7}_{nf}) + s_4(f) \cdot \iota_{I_P}(\tilde{P}^{6}_{nf}) \\
\iota_{I_P}(\tilde{P}^{6}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{12}_{nf}) + s_5(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{14}_{nf}) \\
\iota_{I_P}(\tilde{P}^{7}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{8}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{5}_{nf}) + s_4(f) \cdot \iota_{I_P}(\tilde{P}^{12}_{nf}) \\
\iota_{I_P}(\tilde{P}^{8}_{nf}) &= s_4(f) \cdot \iota_{I_P}(\tilde{P}^{10}_{nf}) \\
\iota_{I_P}(\tilde{P}^{10}_{nf}) &= s_5(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{15}_{nf}) \\
\iota_{I_P}(\tilde{P}^{12}_{nf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{10}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{6}_{nf}) + s_5(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{13}_{nf}) \\
\iota_{I_P}(\tilde{P}^{13}_{nf}) &= r_8(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{17}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{15}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{14}_{nf}) \\
\iota_{I_P}(\tilde{P}^{14}_{nf}) &= r_8(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}_{\bar{n}}) + \iota \cdot \iota_{I_P}(\tilde{P}^{13}_{fn}) \\
\iota_{I_P}(\tilde{P}^{15}_{nf}) &= r_8(\mathrm{ack}) \cdot \iota_{I_P}(\tilde{P}^{18}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{14}_{nf}) \\
\iota_{I_P}(\tilde{P}^{17}_{nf}) &= \sum\nolimits_{g \in D} r_1(g) \cdot \iota_{I_P}(\tilde{P}^{20}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{18}_{nf}) + \iota \cdot \iota_{I_P}(\tilde{P}_{\bar{n}}) \\
\iota_{I_P}(\tilde{P}^{18}_{nf}) &= \sum\nolimits_{g \in D} r_1(g) \cdot \iota_{I_P}(\tilde{P}^{21}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}_{\bar{n}}) \\
\iota_{I_P}(\tilde{P}^{20}_{ngf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{21}_{ngf}) + \iota \cdot \iota_{I_P}(\tilde{P}^{1}_{\bar{n}g}) \\
\iota_{I_P}(\tilde{P}^{21}_{ngf}) &= \iota \cdot \iota_{I_P}(\tilde{P}^{1}_{\bar{n}g})
\end{aligned}$$

5.5 Module Q

5.5.1 Abbreviations

Let $n \in B, \bar{n} = 1 - n$,
$L_n = (i \cdot s_7(n) + i \cdot s_7(\perp) + i) \cdot L$,
$As(0) = As_0 \cdot As$,
$As(1) = As$,
$Ar(0) = Ar$,
$Ar(1) = Ar_1 \cdot Ar$.

We see that $Q \equiv \iota_{I_Q}(Q_0)$.

$$
\begin{array}{rcl}
Q_n & = & \partial_{H_Q}(As(\bar{n}) \parallel L \parallel Ar(n)) \\
Q_n^1 & = & \partial_{H_Q}(As(n) \parallel L \parallel Ar(n)) \\
Q_n^2 & = & \partial_{H_Q}(As(\bar{n}) \parallel L_{\bar{n}} \parallel Ar(n)) \\
Q_n^3 & = & \partial_{H_Q}(As(n) \parallel L_n \parallel Ar(n)) \\
Q_n^4 & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(\bar{n}) \cdot L \parallel Ar(n)) \\
Q_n^5 & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(\perp) \cdot L \parallel Ar(n)) \\
Q_n^6 & = & \partial_{H_Q}(As(n) \parallel s_7(n) \cdot L \parallel Ar(n)) \\
Q_n^7 & = & \partial_{H_Q}(As(n) \parallel s_7(\perp) \cdot L \parallel Ar(n)) \\
Q_n^8 & = & \partial_{H_Q}(As(n) \parallel s_7(\bar{n}) \cdot L \parallel Ar(n)) \\
Q_n^9 & = & \partial_{H_Q}(As(n) \parallel L_{\bar{n}} \parallel Ar(n)) \\
Q_n^{10} & = & \partial_{H_Q}(As(n) \parallel L \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
Q_n^{11} & = & \partial_{H_Q}(As(n) \parallel L_n \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
Q_n^{13} & = & \partial_{H_Q}(As(n) \parallel s_7(n) \cdot L \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
Q_n^{14} & = & \partial_{H_Q}(As(n) \parallel s_7(\perp) \cdot L \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n}))
\end{array}
$$

$$
\begin{array}{rcl}
R_n^1 & = & \partial_{H_Q}(As(\bar{n}) \parallel L_n \parallel Ar(n)) \\
R_n^2 & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(n) \cdot L \parallel Ar(n)) \\
R_n^3 & = & \partial_{H_Q}(As(\bar{n}) \parallel L \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
R_n^4 & = & \partial_{H_Q}(As(\bar{n}) \parallel L_n \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
R_n^5 & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(n) \cdot L \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
R_n^6 & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(\perp) \cdot L \parallel s_8(\mathrm{ack}) \cdot Ar(\bar{n})) \\
R_n^7 & = & \partial_{H_Q}(As(\bar{n}) \parallel L \parallel Ar(n)) \\
R_n^8 & = & \partial_{H_Q}(As(\bar{n}) \parallel L_{\bar{n}} \parallel Ar(n)) \\
R_n^9 & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(\bar{n}) \cdot L \parallel Ar(n)) \\
R_n^{10} & = & \partial_{H_Q}(As(\bar{n}) \parallel s_7(\perp) \cdot L \parallel Ar(n))
\end{array}
$$

5.5.2 Linearisation

In Section 5.5.3, we see that summands $r_5(\text{ack}) \cdot R_n^k$ are omitted. We therefore do not linearise process terms R_n^k.

It is straightforward to derive the following equations.

$$
\begin{array}{rcl}
Q_n & = & r_5(\text{ack}) \cdot Q_n^1 + c_6(\bar{n}) \cdot Q_n^2 \\
Q_n^1 & = & r_5(\text{ack}) \cdot R_n^7 + c_6(n) \cdot Q_n^3 \\
Q_n^2 & = & r_5(\text{ack}) \cdot Q_n^9 + i \cdot Q_n^4 + i \cdot Q_n^5 + i \cdot Q_n \\
Q_n^3 & = & r_5(\text{ack}) \cdot R_n^1 + i \cdot Q_n^6 + i \cdot Q_n^7 + i \cdot Q_n^1 \\
Q_n^4 & = & r_5(\text{ack}) \cdot Q_n^8 + c_7(\bar{n}) \cdot Q_n \\
Q_n^5 & = & r_5(\text{ack}) \cdot Q_n^7 + c_7(\bot) \cdot Q_n \\
Q_n^6 & = & r_5(\text{ack}) \cdot R_n^2 + c_7(n) \cdot Q_n^{10} \\
Q_n^7 & = & r_5(\text{ack}) \cdot R_n^{10} + c_7(\bot) \cdot Q_n^1 \\
Q_n^8 & = & r_5(\text{ack}) \cdot R_n^9 + c_7(\bar{n}) \cdot Q_n^1 \\
Q_n^9 & = & r_5(\text{ack}) \cdot R_n^8 + i \cdot Q_n^8 + i \cdot Q_n^7 + i \cdot Q_n^1 \\
Q_n^{10} & = & r_5(\text{ack}) \cdot R_n^3 + c_6(n) \cdot Q_n^{11} + s_8(\text{ack}) \cdot Q_{\bar{n}} \\
Q_n^{11} & = & r_5(\text{ack}) \cdot R_n^4 + i \cdot Q_n^{13} + i \cdot Q_n^{14} + i \cdot Q_n^{10} + s_8(\text{ack}) \cdot Q_{\bar{n}}^2 \\
Q_n^{13} & = & r_5(\text{ack}) \cdot R_n^5 + s_8(\text{ack}) \cdot Q_{\bar{n}}^4 \\
Q_n^{14} & = & r_5(\text{ack}) \cdot R_n^6 + s_8(\text{ack}) \cdot Q_{\bar{n}}^5
\end{array}
$$

5.5.3 Getting rid of redundant summands

All traces of the localization of Q in context $\partial_H(P \parallel Q)$ satisfy property S:

$$S(\sigma): \quad \text{for all } u, v \text{ such that } \sigma = uv \text{ it holds that } \; N(r_5(\text{ack}), u) \leq N(s_8(\text{ack}), u) + 1$$

As in Section 5.4.3, this statement is based on meta-knowledge.

Now let

$$Z = \{\sigma \in A_\iota{}^* \;\mid\; S(\sigma)\}$$

Then $tr \circ \nu_{\alpha(Q)} \circ \partial_H(\bar{P}_0 \parallel Q) \subseteq Z$ and by Theorem 3.5

$$\partial_H(\bar{P}_0 \parallel Q) = \partial_H(\bar{P}_0 \parallel \nabla_Z(Q))$$

Because Z is such that for the restriction of Q to Z only actions outside I_Q matter, we have

$$\nabla_Z(Q) = \nabla_Z \circ \iota_{I_Q}(Q_0) = \iota_{I_Q} \circ \nabla_Z(Q_0)$$

For all abbreviations X on the left-hand sides of the equations in Section 5.5.2, we write $\tilde{X}$ to abbreviate $\nabla_Z(X)$. The reader is invited to check that the FLS below can be derived using the axioms for restriction.

$$
\begin{array}{rcl}
\tilde{Q}_n & = & r_5(\text{ack}) \cdot \tilde{Q}_n^1 + c_6(\bar{n}) \cdot \tilde{Q}_n^2
\end{array}
$$

$$
\begin{array}{rcl}
\tilde{Q}_n^1 & = & c_6(n) \cdot \tilde{Q}_n^3 \\
\tilde{Q}_n^2 & = & r_5(\text{ack}) \cdot \tilde{Q}_n^9 + i \cdot \tilde{Q}_n^4 + i \cdot \tilde{Q}_n^5 + i \cdot \tilde{Q}_n \\
\tilde{Q}_n^3 & = & i \cdot \tilde{Q}_n^6 + i \cdot \tilde{Q}_n^7 + i \cdot \tilde{Q}_n^1 \\
\tilde{Q}_n^4 & = & r_5(\text{ack}) \cdot \tilde{Q}_n^8 + c_7(\bar{n}) \cdot \tilde{Q}_n \\
\tilde{Q}_n^5 & = & r_5(\text{ack}) \cdot \tilde{Q}_n^7 + c_7(\perp) \cdot \tilde{Q}_n \\
\tilde{Q}_n^6 & = & c_7(n) \cdot \tilde{Q}_n^{10} \\
\tilde{Q}_n^7 & = & c_7(\perp) \cdot \tilde{Q}_n^1 \\
\tilde{Q}_n^8 & = & c_7(\bar{n}) \cdot \tilde{Q}_n^1 \\
\tilde{Q}_n^9 & = & i \cdot \tilde{Q}_n^8 + i \cdot \tilde{Q}_n^7 + i \cdot \tilde{Q}_n^1 \\
\tilde{Q}_n^{10} & = & c_6(n) \cdot \tilde{Q}_n^{11} + s_8(\text{ack}) \cdot \tilde{Q}_{\bar{n}} \\
\tilde{Q}_n^{11} & = & i \cdot \tilde{Q}_n^{13} + i \cdot \tilde{Q}_n^{14} + i \cdot \tilde{Q}_n^{10} + s_8(\text{ack}) \cdot \tilde{Q}_{\bar{n}}^2 \\
\tilde{Q}_n^{13} & = & s_8(\text{ack}) \cdot \tilde{Q}_{\bar{n}}^4 \\
\tilde{Q}_n^{14} & = & s_8(\text{ack}) \cdot \tilde{Q}_{\bar{n}}^5
\end{array}
$$

5.5.4 Abstracting from Internal Activity

We found that $Q = \iota_{I_Q}(\tilde{Q}_0)$. In this section we reduce $\iota_{I_Q}(\tilde{Q}_0)$. In the scope of ι_{I_Q}, $\tilde{Q}_n^1$, $\tilde{Q}_n^3$ and $\tilde{Q}_n^7$ form a cluster with exit $i \cdot \tilde{Q}_n^6$.

$$
\begin{array}{rcl}
\iota_{I_Q}(\tilde{Q}_n) & = & r_5(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}_n^1) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2) \\
& \stackrel{\text{CFAR}^\iota}{=} & r_5(\text{ack}) \cdot \iota_{I_Q}(i \cdot \tilde{Q}_n^6) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2) \\
& = & r_5(\text{ack}) \cdot \iota \cdot \iota_{I_Q}(c_7(n) \cdot \tilde{Q}_n^{10}) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2) \\
& = & r_5(\text{ack}) \cdot \iota \cdot \iota \cdot \iota_{I_Q}(\tilde{Q}_n^{10}) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2) \\
& = & r_5(\text{ack}) \cdot \iota \cdot \iota_{I_Q}(\tilde{Q}_n^{10}) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2) \\
& = & r_5(\text{ack}) \cdot \iota_{I_Q}(i \cdot \tilde{Q}_n^{10}) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2)
\end{array}
$$

In a similar way, using CFAR$^\iota$, we derive

$$
\begin{array}{rcl}
\iota_{I_Q}(\tilde{Q}_n^5) & = & r_5(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}_n^7) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n) \\
& = & r_5(\text{ack}) \cdot \iota_{I_Q}(i \cdot \tilde{Q}_n^{10}) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n) \\
\iota_{I_Q}(\tilde{Q}_n^8) & = & \iota \cdot \iota_{I_Q}(i \cdot \tilde{Q}_n^{10}) \\
\iota_{I_Q}(\tilde{Q}_n^9) & = & \iota \cdot \iota_{I_Q}(\tilde{Q}_n^8) + \iota \cdot \iota_{I_Q}(i \cdot \tilde{Q}_n^{10})
\end{array}
$$

Now it is straightforward to derive the following FLS.

$$
\begin{array}{rcl}
\iota_{I_Q}(\tilde{Q}_n) & = & r_5(\text{ack}) \cdot \iota_{I_Q}(i \cdot \tilde{Q}_n^{10}) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^2) \\
\iota_{I_Q}(i \cdot \tilde{Q}_n^{10}) & = & \iota \cdot \iota_{I_Q}(\tilde{Q}_n^{10}) \\
\iota_{I_Q}(\tilde{Q}_n^2) & = & r_5(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}_n^9) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^4) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n^5) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n) \\
\iota_{I_Q}(\tilde{Q}_n^4) & = & r_5(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}_n^8) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n)
\end{array}
$$

$$\begin{aligned}
\iota_{I_Q}(\tilde{Q}^5_n) &= r_5(\text{ack}) \cdot \iota_{I_Q}(i \cdot \tilde{Q}^{10}_n) + \iota \cdot \iota_{I_Q}(\tilde{Q}_n) \\
\iota_{I_Q}(\tilde{Q}^8_n) &= \iota \cdot \iota_{I_Q}(i \cdot \tilde{Q}^{10}_n) \\
\iota_{I_Q}(\tilde{Q}^9_n) &= \iota \cdot \iota_{I_Q}(\tilde{Q}^8_n) + \iota \cdot \iota_{I_Q}(i \cdot \tilde{Q}^{10}_n) \\
\iota_{I_Q}(\tilde{Q}^{10}_n) &= \iota \cdot \iota_{I_Q}(\tilde{Q}^{11}_n) + s_8(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}_{\bar{n}}) \\
\iota_{I_Q}(\tilde{Q}^{11}_n) &= \iota \cdot \iota_{I_Q}(\tilde{Q}^{13}_n) + \iota \cdot \iota_{I_Q}(\tilde{Q}^{14}_n) + \iota \cdot \iota_{I_Q}(\tilde{Q}^{10}_n) + s_8(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}^2_{\bar{n}}) \\
\iota_{I_Q}(\tilde{Q}^{13}_n) &= s_8(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}^4_{\bar{n}}) \\
\iota_{I_Q}(\tilde{Q}^{14}_n) &= s_8(\text{ack}) \cdot \iota_{I_Q}(\tilde{Q}^5_{\bar{n}})
\end{aligned}$$

5.6 Merging the Modules

We combine our results from the previous sections.

$$\begin{aligned}
\iota_I \circ \partial_H(P \parallel Q) &\overset{(Section\ 5.4)}{=} \iota_I \circ \partial_H(\iota_{I_P}(\tilde{P}_0) \parallel Q) \\
&\overset{(Section\ 5.5)}{=} \iota_I \circ \partial_H(\iota_{I_P}(\tilde{P}_0) \parallel \iota_{I_Q}(\tilde{Q}_0))
\end{aligned}$$

In Section 5.6.2, I present an FLS $\mathcal{L}$ of $\partial_H(\iota_{I_P}(\tilde{P}_0) \parallel \iota_{I_Q}(\tilde{Q}_0))$. It is obtained by straightforward but tedious expansion of the merge using the specifications of $\iota_{I_P}(\tilde{P}_0)$ and $\iota_{I_Q}(\tilde{Q}_0)$ derived in previous sections.

Abbreviations for the processes in $P(\mathcal{L})$ are in Section 5.6.1. We see that $X_0 = \partial_H(\iota_{I_P}(\tilde{P}_0) \parallel \iota_{I_Q}(\tilde{Q}_0))$.

5.6.1 Abbreviations

$$\begin{aligned}
X_n &= \partial_H(\iota_{I_P}(\tilde{P}_n\ ) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^1_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^1_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^2_n &= \partial_H(\iota_{I_P}(\tilde{P}_n\ ) \parallel \iota_{I_Q}(\tilde{Q}^2_n)) \\
X^3_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^5_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^4_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^1_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^2_n)) \\
X^5_n &= \partial_H(\iota_{I_P}(\tilde{P}_n\ ) \parallel \iota_{I_Q}(\tilde{Q}^4_n)) \\
X^6_n &= \partial_H(\iota_{I_P}(\tilde{P}_n\ ) \parallel \iota_{I_Q}(\tilde{Q}^5_n)) \\
X^7_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^7_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^8_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^6_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^9_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^5_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^2_n)) \\
X^{10}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^1_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^4_n)) \\
X^{11}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^1_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^5_n)) \\
X^{12}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^8_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^{13}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{12}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X^{14}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^7_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^2_n)) \\
X^{15}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{14}_{nd}) \parallel \iota_{I_Q}(i \cdot \tilde{Q}^{10}_n)) \\
X^{16}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^6_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^2_n))
\end{aligned}$$

$$\begin{aligned}
X^{35}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{12}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^4_n)) \\
X^{36}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{12}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^5_n)) \\
X^{37}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{14}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^{11}_n)) \\
X^{38}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{14}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^8_n)) \\
X^{39}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{15}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^{10}_n)) \\
X^{40}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{15}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^9_n)) \\
X^{41}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{10}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^4_n)) \\
X^{42}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{10}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^5_n)) \\
X^{43}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{17}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}})) \\
X^{44}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{13}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^{11}_n)) \\
X^{45}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{13}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^8_n)) \\
X^{46}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{14}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^{13}_n)) \\
X^{47}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{14}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^{14}_n)) \\
X^{48}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{18}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}})) \\
X^{49}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{15}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^{11}_n)) \\
X^{50}_{nd} &= \partial_H(\iota_{I_P}(\tilde{P}^{15}_{nd}) \parallel \iota_{I_Q}(\tilde{Q}^8_n)) \\
X^{51}_{ned} &= \partial_H(\iota_{I_P}(\tilde{P}^{20}_{ned}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}))
\end{aligned}$$

$$
\begin{array}{lll}
X_{nd}^{17} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^5) \parallel \iota_{I_Q}(\tilde{Q}_n^4)) \\
X_{nd}^{18} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^5) \parallel \iota_{I_Q}(\tilde{Q}_n^5)) \\
X_{nd}^{19} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{10}) \parallel \iota_{I_Q}(\tilde{Q}_n)) \\
X_{nd}^{20} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^8) \parallel \iota_{I_Q}(\tilde{Q}_n^2)) \\
X_{nd}^{21} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{13}) \parallel \iota_{I_Q}(i \cdot \tilde{Q}_n^{10})) \\
X_{nd}^{22} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{12}) \parallel \iota_{I_Q}(\tilde{Q}_n^2)) \\
X_{nd}^{23} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^7) \parallel \iota_{I_Q}(\tilde{Q}_n^4)) \\
X_{nd}^{24} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^7) \parallel \iota_{I_Q}(\tilde{Q}_n^5)) \\
X_{nd}^{25} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{14}) \parallel \iota_{I_Q}(\tilde{Q}_n^{10})) \\
X_{nd}^{26} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{14}) \parallel \iota_{I_Q}(\tilde{Q}_n^9)) \\
X_{nd}^{27} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^6) \parallel \iota_{I_Q}(\tilde{Q}_n^4)) \\
X_{nd}^{28} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^6) \parallel \iota_{I_Q}(\tilde{Q}_n^5)) \\
X_{nd}^{29} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{15}) \parallel \iota_{I_Q}(i \cdot \tilde{Q}_n^{10})) \\
X_{nd}^{30} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{10}) \parallel \iota_{I_Q}(\tilde{Q}_n^2)) \\
X_{nd}^{31} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^8) \parallel \iota_{I_Q}(\tilde{Q}_n^4)) \\
X_{nd}^{32} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^8) \parallel \iota_{I_Q}(\tilde{Q}_n^5)) \\
X_{nd}^{33} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{13}) \parallel \iota_{I_Q}(\tilde{Q}_n^{10})) \\
X_{nd}^{34} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{13}) \parallel \iota_{I_Q}(\tilde{Q}_n^9))
\end{array}
$$

$$
\begin{array}{lll}
X_{nd}^{52} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{17}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^2)) \\
X_{nd}^{53} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{13}) \parallel \iota_{I_Q}(\tilde{Q}_n^{13})) \\
X_{nd}^{54} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{13}) \parallel \iota_{I_Q}(\tilde{Q}_n^{14})) \\
X_{nd}^{55} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{18}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^2)) \\
X_{nd}^{56} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{15}) \parallel \iota_{I_Q}(\tilde{Q}_n^{13})) \\
X_{nd}^{57} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{15}) \parallel \iota_{I_Q}(\tilde{Q}_n^{14})) \\
X_{ned}^{58} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{21}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}})) \\
X_{ned}^{59} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{20}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^2)) \\
X_{nd}^{60} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{17}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^4)) \\
X_{nd}^{61} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{17}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^5)) \\
X_{ned}^{62} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{21}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^2)) \\
X_{nd}^{63} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{18}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^4)) \\
X_{nd}^{64} & = & \partial_H(\iota_{I_P}(\tilde{P}_{nd}^{18}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^5)) \\
X_{ned}^{65} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{20}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^4)) \\
X_{ned}^{66} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{20}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^5)) \\
X_{ned}^{67} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{21}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^4)) \\
X_{ned}^{68} & = & \partial_H(\iota_{I_P}(\tilde{P}_{ned}^{21}) \parallel \iota_{I_Q}(\tilde{Q}_{\bar{n}}^5))
\end{array}
$$

5.6.2 Linearisation

$$
\begin{array}{lll}
X_n & = & \sum_{d \in D} r_1(d) \cdot X_{nd}^1 + \iota \cdot X_n^2 \\
X_{nd}^1 & = & i \cdot X_{nd}^3 + \iota \cdot X_{nd}^4 \\
X_n^2 & = & \sum_{d \in D} r_1(d) \cdot X_{nd}^4 + \iota \cdot X_n^5 + \iota \cdot X_n^6 + \iota \cdot X_n \\
X_{nd}^3 & = & \iota \cdot X_{nd}^7 + s_4(d) \cdot X_{nd}^8 + \iota \cdot X_{nd}^9 \\
X_{nd}^4 & = & \iota \cdot X_{nd}^9 + \iota \cdot X_{nd}^{10} + \iota \cdot X_{nd}^{11} + \iota \cdot X_{nd}^1 \\
X_n^5 & = & \sum_{d \in D} r_1(d) \cdot X_{nd}^{10} + \iota \cdot X_n \\
X_n^6 & = & \sum_{d \in D} r_1(d) \cdot X_{nd}^{11} + \iota \cdot X_n \\
X_{nd}^7 & = & \iota \cdot X_{nd}^{12} + \iota \cdot X_{nd}^3 + s_4(d) \cdot X_{nd}^{13} + \iota \cdot X_{nd}^{14} \\
X_{nd}^8 & = & \iota \cdot X_{nd}^{13} + c_5(\text{ack}) \cdot X_{nd}^{15} + \iota \cdot X_{nd}^{16} \\
X_{nd}^9 & = & \iota \cdot X_{nd}^{14} + s_4(d) \cdot X_{nd}^{16} + \iota \cdot X_{nd}^{17} + \iota \cdot X_{nd}^{18} + \iota \cdot X_{nd}^3 \\
X_{nd}^{10} & = & \iota \cdot X_{nd}^{17} + \iota \cdot X_{nd}^1 \\
X_{nd}^{11} & = & \iota \cdot X_{nd}^{18} + \iota \cdot X_{nd}^1 \\
X_{nd}^{12} & = & s_4(d) \cdot X_{nd}^{19} + \iota \cdot X_{nd}^{20} \\
X_{nd}^{13} & = & \iota \cdot X_{nd}^{19} + \iota \cdot X_{nd}^8 + c_5(\text{ack}) \cdot X_{nd}^{21} + \iota \cdot X_{nd}^{22} \\
X_{nd}^{14} & = & \iota \cdot X_{nd}^{20} + \iota \cdot X_{nd}^9 + s_4(d) \cdot X_{nd}^{22} + \iota \cdot X_{nd}^{23} + \iota \cdot X_{nd}^{24} + \iota \cdot X_{nd}^7 \\
X_{nd}^{15} & = & \iota \cdot X_{nd}^{21} + \iota \cdot X_{nd}^{25} \\
X_{nd}^{16} & = & \iota \cdot X_{nd}^{22} + c_5(\text{ack}) \cdot X_{nd}^{26} + \iota \cdot X_{nd}^{27} + \iota \cdot X_{nd}^{28} + \iota \cdot X_{nd}^8
\end{array}
$$

$$
\begin{aligned}
X_{nd}^{17} &= \iota \cdot X_{nd}^{23} + s_4(d) \cdot X_{nd}^{27} + \iota \cdot X_{nd}^{3} \\
X_{nd}^{18} &= \iota \cdot X_{nd}^{24} + s_4(d) \cdot X_{nd}^{28} + \iota \cdot X_{nd}^{3} \\
X_{nd}^{19} &= c_5(\mathrm{ack}) \cdot X_{nd}^{29} + \iota \cdot X_{nd}^{30} \\
X_{nd}^{20} &= s_4(d) \cdot X_{nd}^{30} + \iota \cdot X_{nd}^{31} + \iota \cdot X_{nd}^{32} + \iota \cdot X_{nd}^{12} \\
X_{nd}^{21} &= \iota \cdot X_{nd}^{29} + \iota \cdot X_{nd}^{15} + \iota \cdot X_{nd}^{33} \\
X_{nd}^{22} &= \iota \cdot X_{nd}^{30} + \iota \cdot X_{nd}^{16} + c_5(\mathrm{ack}) \cdot X_{nd}^{34} + \iota \cdot X_{nd}^{35} + \iota \cdot X_{nd}^{36} + \iota \cdot X_{nd}^{13} \\
X_{nd}^{23} &= \iota \cdot X_{nd}^{31} + \iota \cdot X_{nd}^{17} + s_4(d) \cdot X_{nd}^{35} + \iota \cdot X_{nd}^{7} \\
X_{nd}^{24} &= \iota \cdot X_{nd}^{32} + \iota \cdot X_{nd}^{18} + s_4(d) \cdot X_{nd}^{36} + \iota \cdot X_{nd}^{7} \\
X_{nd}^{25} &= c_8(\mathrm{ack}) \cdot X_{\bar{n}} + \iota \cdot X_{nd}^{33} + \iota \cdot X_{nd}^{37} \\
X_{nd}^{26} &= \iota \cdot X_{nd}^{34} + \iota \cdot X_{nd}^{38} + \iota \cdot X_{nd}^{15} \\
X_{nd}^{27} &= \iota \cdot X_{nd}^{35} + c_5(\mathrm{ack}) \cdot X_{nd}^{38} + \iota \cdot X_{nd}^{8} \\
X_{nd}^{28} &= \iota \cdot X_{nd}^{36} + c_5(\mathrm{ack}) \cdot X_{nd}^{15} + \iota \cdot X_{nd}^{8} \\
X_{nd}^{29} &= \iota \cdot X_{nd}^{15} + \iota \cdot X_{nd}^{39} \\
X_{nd}^{30} &= c_5(\mathrm{ack}) \cdot X_{nd}^{40} + \iota \cdot X_{nd}^{41} + \iota \cdot X_{nd}^{42} + \iota \cdot X_{nd}^{19} \\
X_{nd}^{31} &= s_4(d) \cdot X_{nd}^{41} + \iota \cdot X_{nd}^{12} \\
X_{nd}^{32} &= s_4(d) \cdot X_{nd}^{42} + \iota \cdot X_{nd}^{12} \\
X_{nd}^{33} &= c_8(\mathrm{ack}) \cdot X_{nd}^{43} + \iota \cdot X_{nd}^{39} + \iota \cdot X_{nd}^{25} + \iota \cdot X_{nd}^{44} \\
X_{nd}^{34} &= \iota \cdot X_{nd}^{40} + \iota \cdot X_{nd}^{26} + \iota \cdot X_{nd}^{45} + \iota \cdot X_{nd}^{21} \\
X_{nd}^{35} &= \iota \cdot X_{nd}^{41} + \iota \cdot X_{nd}^{27} + c_5(\mathrm{ack}) \cdot X_{nd}^{45} + \iota \cdot X_{nd}^{13} \\
X_{nd}^{36} &= \iota \cdot X_{nd}^{42} + \iota \cdot X_{nd}^{28} + c_5(\mathrm{ack}) \cdot X_{nd}^{21} + \iota \cdot X_{nd}^{13} \\
X_{nd}^{37} &= c_8(\mathrm{ack}) \cdot X_{\bar{n}}^{2} + \iota \cdot X_{nd}^{44} + \iota \cdot X_{nd}^{46} + \iota \cdot X_{nd}^{47} + \iota \cdot X_{nd}^{25} \\
X_{nd}^{38} &= \iota \cdot X_{nd}^{45} + \iota \cdot X_{nd}^{15} \\
X_{nd}^{39} &= c_8(\mathrm{ack}) \cdot X_{nd}^{48} + \iota \cdot X_{nd}^{25} + \iota \cdot X_{nd}^{49} \\
X_{nd}^{40} &= \iota \cdot X_{nd}^{26} + \iota \cdot X_{nd}^{50} + i\iota \cdot X_{nd}^{29} \\
X_{nd}^{41} &= c_5(\mathrm{ack}) \cdot X_{nd}^{50} + \iota \cdot X_{nd}^{19} \\
X_{nd}^{42} &= c_5(\mathrm{ack}) \cdot X_{nd}^{29} + \iota \cdot X_{nd}^{19} \\
X_{nd}^{43} &= \sum\nolimits_{e \in D} r_1(e) \cdot X_{ned}^{51} + \iota \cdot X_{nd}^{48} + \iota \cdot X_{\bar{n}} + \iota \cdot X_{nd}^{52} \\
X_{nd}^{44} &= c_8(\mathrm{ack}) \cdot X_{nd}^{52} + \iota \cdot X_{nd}^{49} + \iota \cdot X_{nd}^{37} + \iota \cdot X_{nd}^{53} + \iota \cdot X_{nd}^{54} + \iota \cdot X_{nd}^{33} \\
X_{nd}^{45} &= \iota \cdot X_{nd}^{50} + \iota \cdot X_{nd}^{38} + \iota \cdot X_{nd}^{21} \\
X_{nd}^{46} &= c_8(\mathrm{ack}) \cdot X_{\bar{n}}^{5} + \iota \cdot X_{nd}^{53} \\
X_{nd}^{47} &= c_8(\mathrm{ack}) \cdot X_{\bar{n}}^{6} + \iota \cdot X_{nd}^{54} \\
X_{nd}^{48} &= \sum\nolimits_{e \in D} r_1(e) \cdot X_{ned}^{58} + \iota \cdot X_{\bar{n}} + \iota \cdot X_{nd}^{55} \\
X_{nd}^{49} &= c_8(\mathrm{ack}) \cdot X_{nd}^{55} + \iota \cdot X_{nd}^{37} + \iota \cdot X_{nd}^{56} + \iota \cdot X_{nd}^{57} + \iota \cdot X_{nd}^{39} \\
X_{nd}^{50} &= \iota \cdot X_{nd}^{38} + \iota \cdot X_{nd}^{29} \\
X_{ned}^{51} &= \iota \cdot X_{ned}^{58} + \iota \cdot X_{\bar{n}e}^{1} + \iota \cdot X_{ned}^{59} \\
X_{nd}^{52} &= \sum\nolimits_{e \in D} r_1(e) \cdot X_{ned}^{59} + \iota \cdot X_{nd}^{55} + \iota \cdot X_{\bar{n}}^{2} + \iota \cdot X_{nd}^{60} + \iota \cdot X_{nd}^{61} + \iota \cdot X_{nd}^{43} \\
X_{nd}^{53} &= c_8(\mathrm{ack}) \cdot X_{nd}^{60} + \iota \cdot X_{nd}^{56} + \iota \cdot X_{nd}^{46} \\
X_{nd}^{54} &= c_8(\mathrm{ack}) \cdot X_{nd}^{61} + \iota \cdot X_{nd}^{57} + \iota \cdot X_{nd}^{47} \\
X_{nd}^{55} &= \sum\nolimits_{e \in D} r_1(e) \cdot X_{ned}^{62} + \iota \cdot X_{\bar{n}}^{2} + \iota \cdot X_{nd}^{63} + \iota \cdot X_{nd}^{64} + \iota \cdot X_{nd}^{48}
\end{aligned}
$$

$$
\begin{array}{rcl}
X_{nd}^{56} & = & c_8(\text{ack}) \cdot X_{nd}^{63} + \iota \cdot X_{nd}^{46} \\
X_{nd}^{57} & = & c_8(\text{ack}) \cdot X_{nd}^{64} + \iota \cdot X_{nd}^{47} \\
X_{ned}^{58} & = & \iota \cdot X_{\bar{n}e}^{1} + \iota \cdot X_{ned}^{62} \\
X_{ned}^{59} & = & \iota \cdot X_{ned}^{62} + \iota \cdot X_{\bar{n}e}^{4} + \iota \cdot X_{ned}^{65} + \iota \cdot X_{ned}^{66} + \iota \cdot X_{ned}^{51} \\
X_{nd}^{60} & = & \sum_{e \in D} r_1(e) \cdot X_{ned}^{65} + \iota \cdot X_{nd}^{63} + \iota \cdot X_{\bar{n}}^{5} + \iota \cdot X_{nd}^{43} \\
X_{nd}^{61} & = & \sum_{e \in D} r_1(e) \cdot X_{ned}^{66} + \iota \cdot X_{nd}^{64} + \iota \cdot X_{\bar{n}}^{6} + \iota \cdot X_{nd}^{43} \\
X_{ned}^{62} & = & \iota \cdot X_{\bar{n}e}^{4} + \iota \cdot X_{ned}^{67} + \iota \cdot X_{ned}^{68} + \iota \cdot X_{ned}^{58} \\
X_{nd}^{63} & = & \sum_{e \in D} r_1(e) \cdot X_{ned}^{67} + \iota \cdot X_{\bar{n}}^{5} + \iota \cdot X_{nd}^{48} \\
X_{nd}^{64} & = & \sum_{e \in D} r_1(e) \cdot X_{ned}^{68} + \iota \cdot X_{\bar{n}}^{6} + \iota \cdot X_{nd}^{48} \\
X_{ned}^{65} & = & \iota \cdot X_{ned}^{67} + \iota \cdot X_{\bar{n}e}^{10} + \iota \cdot X_{ned}^{51} \\
X_{ned}^{66} & = & \iota \cdot X_{ned}^{68} + \iota \cdot X_{\bar{n}e}^{11} + \iota \cdot X_{ned}^{51} \\
X_{ned}^{67} & = & \iota \cdot X_{\bar{n}e}^{10} + \iota \cdot X_{ned}^{58} \\
X_{ned}^{68} & = & \iota \cdot X_{\bar{n}e}^{11} + \iota \cdot X_{ned}^{58}
\end{array}
$$

5.7 Verification

We will see that

$$[\![CABP]\!] \quad \underline{\leftrightarrow}_{ro} \quad [\![(\iota^* \sum_{d \in D} r_1(d) \cdot \iota \cdot (\iota^* s_4(d)) \cdot \iota)^* \delta]\!]$$

In the previous sections we have seen

$$CABP \quad = \quad \iota_I(X_0)$$

We found an FLS $\mathcal{L}$ of X_0 in Section 5.6. The structure of $\mathcal{L}$ is too obscure to proceed algebraically. In particular, it is not clear what fairness rules we could apply. Therefore we give an interpretation of $\mathcal{L}$ and show

$$[\![\iota_I(X_0)]\!] \quad \underline{\leftrightarrow}_{ro} \quad [\![(\iota^* \sum_{d \in D} r_1(d) \cdot \iota \cdot (\iota^* s_4(d)) \cdot \iota)^* \delta]\!]$$

Definition 5.1 (*Interpretation of* X_0)**.** Let $\hat{.} : D \to \mathbb{N}^+$ be an encoding of data elements to naturals ≥ 1.

The function $[.] : P(\mathcal{L}) \to S$ is defined as follows.

$$
\begin{array}{rcl}
[X_n] & = & (n) \succ\!\prec (0) \succ\!\prec (0) \succ\!\prec (0) \\
[X_n^i] & = & (n) \succ\!\prec (i) \succ\!\prec (0) \succ\!\prec (0) \\
[X_{nd}^i] & = & (n) \succ\!\prec (i) \succ\!\prec (\hat{d}) \succ\!\prec (0) \\
[X_{ned}^i] & = & (n) \succ\!\prec (i) \succ\!\prec (\hat{e}) \succ\!\prec (\hat{d})
\end{array}
$$

Now a frame F and a pointed frame $([X_0], t, F)$ are found following Definition 2.18. By Theorem 2.19 we have $([X_0], t, F) \in [\![X_0]\!]$. By the definition of ι_I we have $([X_0], t, \iota_I(F)) \in [\![\iota_I(X_0)]\!]$. □

Lemma 5.2. *For all $s \in |F|$ it holds that either*
-the first external action on all paths leaving from s is $r_1(d)$ for some $d \in D$, or
-the first external action on all paths leaving from s is $s_4(d)$ for some $d \in D$.

Proof. Easy to check. $\square$

The function $E_F : |F| \to \mathcal{A}$ gives the first external action on some path leaving the state. E_F is a function by Lemma 5.2.

Lemma 5.3. *Every state $s \in |F|$ has exactly one of the following properties P_1-P_4.*

$$
\begin{array}{ll}
P_1: & \exists s' \exists d.\ s \xrightarrow{r_1(d)} s' \\
P_2: & \exists s' \exists d.\ s \xrightarrow{s_4(d)} s' \\
P_3: & \neg P_1(s) \textit{ and } \neg P_2(s) \textit{ and } \exists d.\ E_F(s) = r_1(d) \\
P_4: & \neg P_1(s) \textit{ and } \neg P_2(s) \textit{ and } \exists d.\ E_F(s) = s_4(d)
\end{array}
$$

Proof. Easy to check, tedious to prove. $\square$

Definition 5.4 (*The Frame G*)**.** Consider the following frame G.

$|G| = \{0, 0\rangle\!\!-\!\!\langle\hat{d}, 1\rangle\!\!-\!\!\langle\hat{d}, 1 \mid d \in D\}$.

For all $d \in D$ the following transitions are in G:

$$
\begin{array}{rcl}
0 & \xrightarrow{\iota} & 0 \\
0 & \xrightarrow{r_1(d)} & 0\rangle\!\!-\!\!\langle\hat{d} \\
0\rangle\!\!-\!\!\langle\hat{d} & \xrightarrow{\iota} & 1\rangle\!\!-\!\!\langle\hat{d} \\
1\rangle\!\!-\!\!\langle\hat{d} & \xrightarrow{\iota} & 1\rangle\!\!-\!\!\langle\hat{d} \\
1\rangle\!\!-\!\!\langle\hat{d} & \xrightarrow{s_4(d)} & 1 \\
1 & \xrightarrow{\iota} & 0
\end{array}
$$

There are no other transitions in G. Now let $t \notin |G|$, then $(0, t, G) \in [\![(\iota^* \sum_{d \in D} r_1(d) \cdot \iota \cdot (\iota^* s_4(d)) \cdot \iota)^* \delta]\!]$. The frame G is depicted in Figure 5.2. $\square$

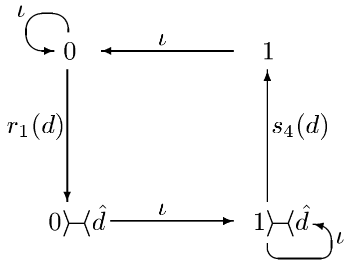


Figure 5.2: The frame G for one datum d

Definition 5.5. The relation $R \subseteq |F| \times |G|$ is the smallest set such that for all $s \in |F|$:

$$
\begin{array}{lcl}
P_1(s) & \Rightarrow & \langle s, 0\rangle \in R \\
P_2(s) & \Rightarrow & \langle s, 1 \succ\!\!\prec \pi^1(\pi^0(s))\rangle \in R \\
P_3(s) & \Rightarrow & \langle s, 1\rangle \in R \\
P_4(s) & \Rightarrow & \langle s, 0 \succ\!\!\prec \pi^1(\pi^0(s))\rangle \in R
\end{array}
$$

Here $\pi^0(x \succ\!\!\prec y) = x$ and $\pi^1(x \succ\!\!\prec y) = y$ for all pairs $x \succ\!\!\prec y$. □

By Lemma 5.3, R relates every state of $\iota_I(F)$ to exactly one state of G. Every state of G is related to at least one state of $\iota_I(F)$.

Theorem 5.6. *R is a rooted orthogonal bisimulation between* $([X_0], t, \iota_I(F))$ *and* $(0, t, G)$.

Corollary 5.7.

$$
[\![CABP]\!] \quad \underline{\leftrightarrow}_{ro} \quad [\![(\iota^* \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot (\iota^* s_4(d)) \cdot \iota)^* \delta]\!]
$$

We see that $[\![CABP]\!]$ is ι-observation bisimilar to the interpretation of B^{14}, so the external behaviour is as intended.

Remark. In our verification of the ABP, we proved the interpretation of *ABP* rooted orthogonal bisimilar to $[\![(\sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota)^* \delta]\!]$. This time, we are, in a way, farther removed from B^{14}. This difference is due to the parallel internal activity. In any state the CABP can do some internal step. The role of parallel internal activity in the outcome of the verification is illustrated by the following example.

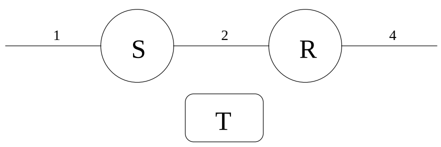


Figure 5.3: A simple communication protocol

Consider the simple communication protocol depicted in Figure 5.3, and specified as follows.

$$
\begin{array}{lcl}
S & = & (\sum_{d \in D} r_1(d) \cdot s_2(d) \cdot r_2(\mathrm{ack}))^* \delta \\
R & = & (\sum_{d \in D} r_2(d) \cdot s_4(d) \cdot s_2(\mathrm{ack}))^* \delta \\
T & = & i^* \delta
\end{array}
$$

Straightforward calculations left to reader yield the following result, with I, H appropriately chosen.

$$
[\![\iota_I \partial_H (S \parallel R \parallel T)]\!] \quad = \quad [\![CABP]\!]
$$

Chapter 6

The PAR Protocol

The architecture of the Positive Acknowledgement and Retransmission (PAR) protocol is depicted in Figure 6.1. I refer to [Vaa90a] for an earlier verification.
The PAR Protocol resembles the alternating bit protocol but a couple of things are different. The receiver has just one (thus *positive*) acknowledgement for the arrival of a frame. The channels in the ABP always manage to send an error message in case data is corrupted. As in the CABP, here channels may fail to do even that.
We do not have parallel internal activity, the sender and the receiver act in response to received data only. This poses a problem, because when either of the channels K and L fails to act upon receiving data, the system will be waiting for nothing to happen, i.e. it will deadlock. In order to avoid this, a timer is added. The timer is started by the sender every time it sends a message to R. The timer is supposed to reactivate the system if it threatens to deadlock, i.e. if no other activity is possible the timer sends a time-out message to the sender. This is achieved by placing the system in the scope of a priority operator and giving the communication of a time-out message over port 7 lower priority than any other action. Following [Vaa90a] we give the communication of the start message highest priority, but this is not necessary for the correctness of the protocol.

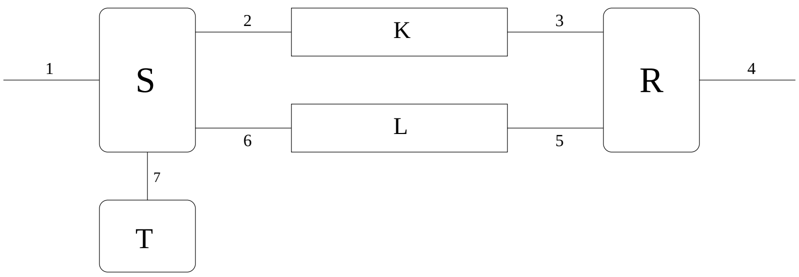


Figure 6.1: Architecture of the PAR protocol

6.1 Specification

We assume a finite data set D, a set of bits $B = \{0, 1\}$, and a set of frames $F = D \times B$. For $n \in B$ we abbreviate $1 - n$ by $\bar{n}$.

The components are specified as follows.

$$\begin{array}{rcl}
S & = & (S_0 \cdot S_1)^* \delta \\
S_n & = & \sum_{d \in D} r_1(d) \cdot s_2(dn) \cdot s_7(\mathrm{st}) \cdot S_{dn} \\
S_{dn} & = & ((r_6(\bot) + \mathrm{to}) \cdot s_2(dn) \cdot \mathrm{st})^* r_6(\mathrm{ack}) \\
T & = & (r_7(\mathrm{st}) \cdot (r_7(\mathrm{st})^* s_7(\mathrm{to})))^* \delta \\
K & = & (\sum_{x \in F} r_2(x) \cdot (i \cdot s_3(x) + i \cdot s_3(\bot) + i))^* \delta \\
L & = & (r_5(\mathrm{ack}) \cdot (i \cdot s_6(\mathrm{ack}) + i \cdot s_6(\bot) + i))^* \delta \\
R & = & (R_0 \cdot R_1)^* \delta \\
R_n & = & (r_3(\bot) + \sum_{d \in D} r_3(d\bar{n}) \cdot s_5(\mathrm{ack}))^* \sum_{d \in D} r_3(dn) \cdot s_4(d) \cdot s_5(\mathrm{ack})
\end{array}$$

To simplify derivations I take components S and T together. It is easy to derive that the process $\partial_{\{s_7(\mathrm{st}), s_7(\mathrm{to}), r_7(\mathrm{st}), r_7(\mathrm{to})\}}(S \parallel T)$ is equal to S^t specified below.

$$\begin{array}{rcl}
S^t & = & (S_0^t \cdot S_1^t)^* \delta \\
S_n^t & = & \sum_{d \in D} r_1(d) \cdot s_2(dn) \cdot c_7(\mathrm{st}) \cdot S_{dn}^t \\
S_{dn}^t & = & ((r_6(\bot) + c_7(\mathrm{to})) \cdot s_2(dn) \cdot c_7(\mathrm{st}))^* r_6(\mathrm{ack})
\end{array}$$

A convenient renaming of c_7 actions gives us

$$\begin{array}{rcl}
S_n^t & = & \sum_{d \in D} r_1(d) \cdot s_2(dn) \cdot \mathrm{st} \cdot S_{dn}^t \\
S_{dn}^t & = & ((r_6(\bot) + \mathrm{to}) \cdot s_2(dn) \cdot \mathrm{st})^* r_6(\mathrm{ack})
\end{array}$$

Now let

$$\begin{array}{rcl}
H & = & \{s_m(x), r_m(x) \mid m \in \{2,3,5,6\},\ x \in \{\mathrm{ack}, \bot\} \cup F\} \\
I & = & \{\mathrm{to}, \mathrm{st}, i\} \cup \{c_m(x) \mid m \in \{2,3,5,6\},\ x \in \{\mathrm{ack}, \bot\} \cup F\}
\end{array}$$

and $to < a < st$ for all atomic actions a other than to and st, then

$$PAR = \iota_I \circ \theta \circ \partial_H(S^t \parallel K \parallel R \parallel L)$$

6.2 Linearisation

Notation: $[p]$ for $\theta \circ \partial_H(p)$. I use the following abbreviations.

$$\begin{array}{rcl}
X & = & [S^t \parallel K \parallel R \parallel L] \\
X_d^0 & = & [s_2(d0) \cdot \mathrm{st} \cdot S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R \parallel L] \\
X_d^1 & = & [S_{d0}^t \cdot S_1^t S^t \parallel (i \cdot s_3(d0) + i \cdot s_3(\bot) + i) \cdot K \parallel R \parallel L] \\
X_d^2 & = & [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel (i \cdot s_6(\mathrm{ack}) + i \cdot s_6(\bot) + i) \cdot L] \\
X_d^3 & = & [s_2(d0) \cdot \mathrm{st} \cdot S_{dn}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel L]
\end{array}$$

$$
\begin{array}{rcl}
X_d^4 & = & [S_{d0}^t \cdot S_1^t S^t \parallel (i \cdot s_3(d0) + i \cdot s_3(\perp) + i) \cdot K \parallel R_1 R \parallel L] \\
Y & = & [S_1^t S^t \parallel K \parallel R_1 R \parallel L] \\
Y_d^1 & = & [S_{d1}^t \cdot S^t \parallel (i \cdot s_3(d1) + i \cdot s_3(\perp) + i) \cdot K \parallel R_1 R \parallel L] \\
Y_d^2 & = & [S_{d1}^t \cdot S^t \parallel K \parallel R \parallel (i \cdot s_6(\mathrm{ack}) + i \cdot s_6(\perp) + i) \cdot L] \\
Y_d^4 & = & [S_{d1}^t \cdot S^t \parallel (i \cdot s_3(d1) + i \cdot s_3(\perp) + i) \cdot K \parallel R \parallel L]
\end{array}
$$

Driving the composed operator $\theta \circ \partial_H$ inwards we leave out all summands that are blocked by either the encapsulation or the priority operator. In particular this means that, in the presence of alternatives, summands starting with a time-out action are renamed into δ, and if summands start with a start action, all other summands are renamed into δ.

$$
\begin{array}{rcl}
X & = & \sum_{d \in D} r_1(d) \cdot X_d^0 \\
X_d^0 & = & c_2(d0) \cdot [\mathrm{st} \cdot S_{d0} \cdot S_1^t S^t \parallel (i \cdot s_3(d0) + i \cdot s_3(\perp) + i) \cdot K \parallel R \parallel L] \\
& = & c_2(d0) \cdot \mathrm{st} \cdot X_d^1 \\
X_d^1 & = & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel s_3(d0) \cdot K \parallel R \parallel L] \ + \\
& & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel s_3(\perp) \cdot K \parallel R \parallel L] \ + \\
& & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R \parallel L] \\
& = & i \cdot c_3(d0) \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel s_4(d) \cdot s_5(\mathrm{ack}) \cdot R_1 R \parallel L] \ + \\
& & i \cdot c_3(\perp) \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R \parallel L] \ + \\
& & i \cdot \mathrm{to} \cdot X_d^0 \\
& = & i \cdot c_3(d0) \cdot s_4(d) \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel s_5(\mathrm{ack}) \cdot R_1 R \parallel L] \ + \\
& & i \cdot c_3(\perp) \cdot \mathrm{to} \cdot X_d^0 \ + \\
& & i \cdot \mathrm{to} \cdot X_d^0 \\
& = & i \cdot c_3(d0) \cdot s_4(d) \cdot s_5(\mathrm{ack}) \cdot X_d^2 + (i \cdot c_3(\perp) + i) \cdot \mathrm{to} \cdot X_d^0 \\
X_d^2 & = & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel s_6(\mathrm{ack}) \cdot L] \ + \\
& & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel s_6(\perp) \cdot L] \ + \\
& & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel L] \\
& = & i \cdot c_6(\mathrm{ack}) \cdot Y \ + \\
& & i \cdot c_6(\perp) \cdot X_d^3 \ + \\
& & i \cdot \mathrm{to} \cdot X_d^3 \\
X_d^3 & = & c_2(d0) \cdot [st \cdot S_{d0}^t \cdot S_1^t S^t \parallel (i \cdot s_3(d0) + i \cdot s_3(\perp) + i) \cdot K \parallel R_1 R \parallel L] \\
& = & c_2(d0) \cdot \mathrm{st} \cdot X_d^4 \\
X_d^4 & = & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel s_3(d0) \cdot K \parallel R_1 R \parallel L] \ + \\
& & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel s_3(\perp) \cdot K \parallel R_1 R \parallel L] \ + \\
& & i \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel L] \\
& = & i \cdot c_3(d0) \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel s_5(\mathrm{ack}) \cdot R_1 R \parallel L] \ + \\
& & i \cdot c_3(\perp) \cdot [S_{d0}^t \cdot S_1^t S^t \parallel K \parallel R_1 R \parallel L] \ + \\
& & i \cdot \mathrm{to} \cdot X_d^3
\end{array}
$$

$$
\begin{aligned}
&= i \cdot c_3(d0) \cdot c_5(\text{ack}) \cdot X_d^2 \ + \\
&\quad\ i \cdot c_3(\perp) \cdot \text{to} \cdot X_d^3 \ + \\
&\quad\ i \cdot \text{to} \cdot X_d^3 \\
&= i \cdot c_3(d0) \cdot c_5(\text{ack}) \cdot X_d^2 + (i \cdot c_3(\perp) + i) \cdot \text{to} \cdot X_d^3
\end{aligned}
$$

After elimination of process names X_d^0 and X_d^3 from the equations for X we get

$$
\begin{aligned}
X &= \textstyle\sum_{d \in D} r_1(d) \cdot c_2(d0) \cdot \text{st} \cdot X_d^1 \\
X_d^1 &= (i \cdot c_3(\perp) + i) \cdot \text{to} \cdot c_2(d0) \cdot \text{st} \cdot X_d^1 \ + \ i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(\text{ack}) \cdot X_d^2 \\
X_d^2 &= (i \cdot c_6(\perp) + i \cdot \text{to}) \cdot c_2(d0) \cdot \text{st} \cdot X_d^4 \ + \ i \cdot c_6(\text{ack}) \cdot Y \\
X_d^4 &= (i \cdot c_3(\perp) + i) \cdot \text{to} \cdot c_2(d0) \cdot \text{st} \cdot X_d^4 \ + \ i \cdot c_3(d0) \cdot c_5(\text{ack}) \cdot X_d^2
\end{aligned}
$$

In a similar way we can derive

$$
\begin{aligned}
Y &= \textstyle\sum_{d \in D} r_1(e) \cdot c_2(e1) \cdot \text{st} \cdot Y_e^1 \\
Y_e^1 &= (i \cdot c_3(\perp) + i) \cdot \text{to} \cdot c_2(e1) \cdot \text{st} \cdot Y_e^1 \ + \ i \cdot c_3(e1) \cdot s_4(e) \cdot c_5(\text{ack}) \cdot Y_e^2 \\
Y_e^2 &= (i \cdot c_6(\perp) + i \cdot \text{to}) \cdot c_2(e1) \cdot \text{st} \cdot Y_e^4 \ + \ i \cdot c_6(\text{ack}) \cdot X \\
Y_e^4 &= (i \cdot c_3(\perp) + i) \cdot \text{to} \cdot c_2(e1) \cdot \text{st} \cdot Y_e^4 \ + \ i \cdot c_3(e1) \cdot c_5(\text{ack}) \cdot Y_e^2
\end{aligned}
$$

6.3 Abstracting from Internal Activity

Using the equations from the previous section and the fairnessrule F from Section 2.4, we derive

$$
\begin{aligned}
\iota_I(X) &= \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot \iota \cdot \iota_I(X_d^1) \\
\iota \cdot \iota_I(X_d^1) &= \iota \cdot \iota_I(i \cdot c_3(d0) \cdot s_4(d) \cdot c_5(\text{ack}) \cdot X_d^2) \\
&= \iota \cdot \iota \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota_I(X_d^2)
\end{aligned}
$$

We see that X_d^2 and X_d^4 are part of a conservative cluster of I, in the sense of Section 2.4. Therefore we can apply the rule CFAR$^\iota$. The only exit is $i \cdot c_6(\text{ack}) \cdot Y$.

$$
\begin{aligned}
\iota \cdot \iota_I(X_d^2) &= \iota \cdot \iota_I(i \cdot c_6(\text{ack}) \cdot Y) \\
&= \iota \cdot \iota \cdot \iota \cdot \iota_I(Y)
\end{aligned}
$$

Putting things together we get

$$
\iota_I(X) = \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot \iota \cdot \iota \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota \cdot \iota \cdot \iota_I(Y)
$$

Compressing ι-steps yields

$$
\iota_I(X) = \textstyle\sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota \cdot \iota_I(Y)
$$

In a similar way we derive

$$\iota_I(Y) = \sum_{e \in D} r_1(e) \cdot \iota \cdot s_4(e) \cdot \iota \cdot \iota_I(X)$$

Eliminating $\iota_I(Y)$ we get

$$\iota_I(X) = \sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota \cdot \sum_{e \in D} r_1(e) \cdot \iota \cdot s_4(e) \cdot \iota \cdot \iota_I(X)$$

This equation captures an FLS interpretation of which is bisimilar to the interpretation of

$$(\sum_{d \in D} r_1(d) \cdot \iota \cdot s_4(d) \cdot \iota \cdot)^* \delta$$

Chapter 7

Conclusion

We have seen that the ABP and the PAR Protocol have relatively straightforward verifications in ACP_ι^*. Also, the length of the proofs is comparable to the length of their counterparts in ACP_τ. We proved the processes *ABP* and *PAR* rooted orthogonal bisimilar to

(1) $\quad (\sum_{d\in D} r_1(d)\cdot\iota\cdot s_4(d)\cdot\iota)^*\delta$

This process is not rooted orthogonal bisimilar to the standard specification of a one-directional, one-element buffer:

(2) $\quad (\sum_{d\in D} r_1(d)\cdot s_4(d))^*\delta$

It could be argued that (1) is a better specification of a buffer than (2), but in any case the external behaviour of the processes *ABP* and *PAR* is as intended.
It should be noted that we did not *derive* *ABP* and *PAR* equal to (1). This derivation is possible if we adopt the recursive specification principle RSP^* (cf. [BBP94]).

The situation for the CABP is slightly more complex. We proved the process *CABP* rooted orthogonal bisimilar to

(3) $\quad (\iota^*\sum_{d\in D} r_1(d)\cdot\iota\cdot(\iota^* s_4(d))\cdot\iota)^*\delta$

This outcome reflects the parallel internal activity that is present in the CABP. We see that externally the process *CABP* behaves as required.
The verification itself is less satisfying; on the level of the modules we can do fewer reductions than in the verification in ACP_τ, and therefore we are stuck with longer specifications. Moreover, the derivation of the final FLS in Section 5.6.2, was not fully algebraic, as remarked in Section 5.4.3.
It seems that we will have to accept that derivations can be longer. As a recommendation for future work I would suggest a verification of the CABP with another division of the components in modules.

The linear specification format defined in Section 2.3 turned out to be very useful. We were able to avoid the use of recursive specification principles in the verifications.

Bibliography

[Bae90] J.C.M. Baeten (ed). *Applications of Process Algebra.* Cambridge Tracts in Theoretical Computer Science 17. Cambridge University Press, 1990.

[BBK86] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fund. Inf.* IX, pp.127-168, 1986.

[BBK87] J.C.M. Baeten, J.A. Bergstra and J.W. Klop. Conditional axioms and α/β-calculus in process algebra. In *Proceedings of the IFIP Conference on Formal Description of Programming Concepts III, Ebberup 1986*, ed. M. Wirsing. North-Holland, Amsterdam, pp.77-103, 1987.

[BBP94] J.A. Bergstra, I. Bethke and A. Ponse. Process algebra with iteration and nesting. *The Computer Journal* 37(4), pp.243-258, 1994.

[BG93] M. Bezem and J.F. Groote. A formal verification of the alternating bit protocol in the Calculus of Contructions. Logic Group Preprint Series No. 88, Utrecht University, 1993.

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

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

[BP94] J.A. Bergstra and A. Ponse. Frame algebra with synchronous communication. In *Working Papers of the International Workshop on Information Systems - Correctness and Reusability IS-CORE'94*, ed. R.J. Wieringa and R.B. Feenstra. Free University, Amsterdam, 1994.

[BP95] J.A. Bergstra and A. Ponse. Frame-Based Process Logics. In *Modal Logic and Process Algebra - A Bisimulation Perspective*, ed. A. Ponse, M. de Rijke and Y. Venema. CSLI, 1995.

[BP98] J.A. Bergstra and A. Ponse. Process Algebra with Iota. Technical Report, University of Amsterdam. In preparation.

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

[GP94] J.F. Groote and A. Ponse. Syntax and semantics of μCRL. In A. Ponse, C. Verhoef and S.F.M. van Vlijmen, eds, *Algebra of Communicating Processes, Utrecht 1994, Workshops in Computing*, pp.26-62, Springer Verlag, 1994.

[GS95] J.F. Groote and J. Springintveld. Focus points and convergent process operators. Technical Report 142, Logic Group Preprint Series, Utrecht University, 1995.

[GV89] R.J Glabbeek and F.W. Vaandrager. Modulair specifications in process algebra - with curious queues. In *Algebraic Methods: Theory, Tools and Applications*, pp. 465-506, ed. M. Wirsing and J.A. Bergstra. LNCS 394, Springer Verlag, 1989.

[GV92] J.F. Groote and F.W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202-206, 1992.

[KM90] C.P.J. Koymans and J.C. Mulder. A modular approach to protocol verification. In [Bae90], pp. 261-306, 1990.

[Mil80] R. Milner. *A Calculus of Communicating Systems.* LNCS, volume 92. Springer Verlag, 1980.

[Vaa90] F.W. Vaandrager. *Algebraic Techniques for Concurrency and Their Application.* PhD Thesis, University of Amsterdam, 1990.

[Vaa90a] F.W. Vaandrager. Two simple protocols. In [Bae90], pp. 23-45, 1990.

[Vaa90b] F.W. Vaandrager. Some observations on redundancy in context. In [Bae90], pp. 237-260, 1990.

[Wam95] J. van Wamel. An algebraic verification of the concurrent alternating bit protocol. In *Verification Techniques for Elementary Data Types and Retransmission Protocols*, PhD thesis. ILLC Dissertation series 1995-6, 1995.

Appendix A

Illustrations

A.1 The Alternating Bit Protocol

Figure A.1 illustrates the linearisation of $\partial_H(S\|K\|R\|L)$.

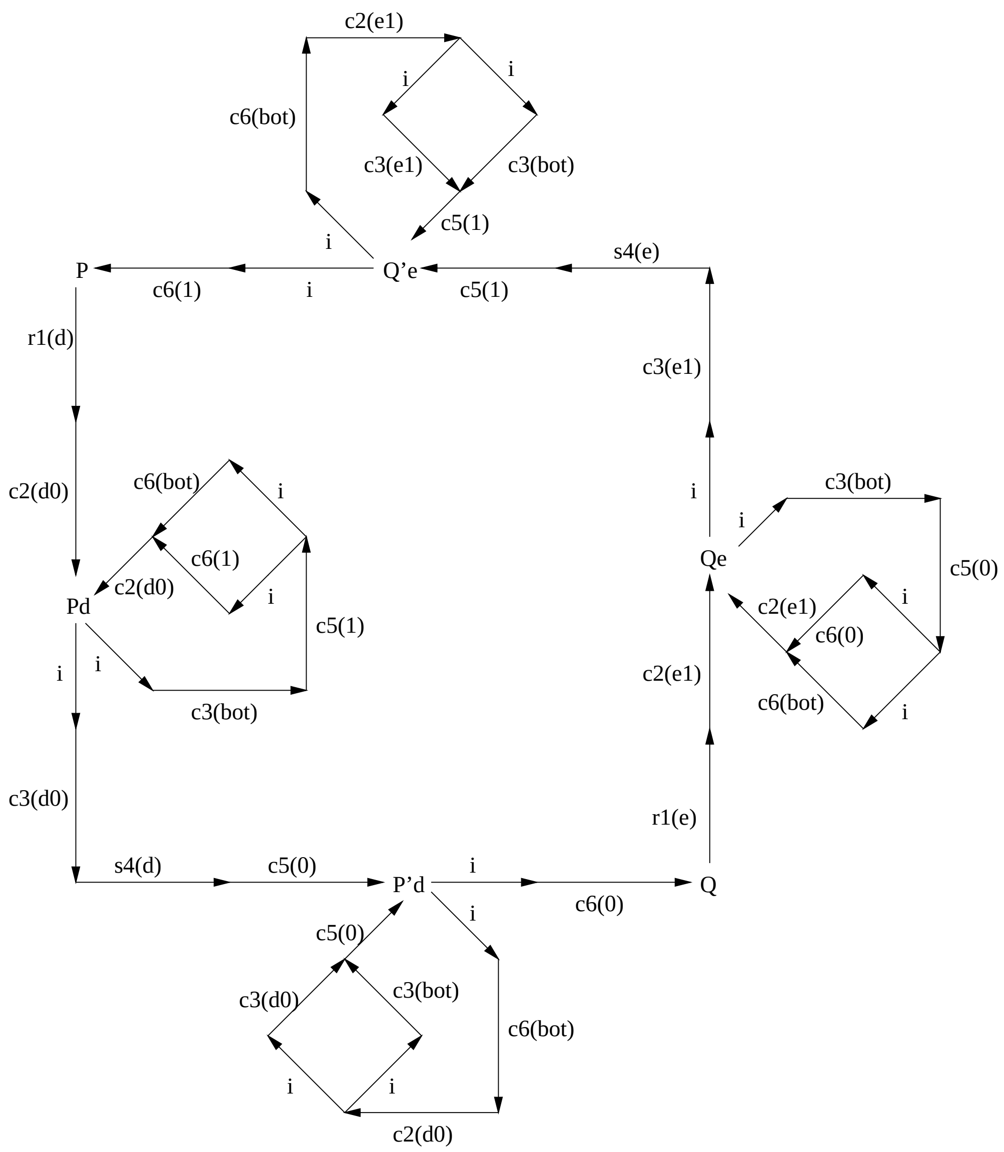


Figure A.1: The linearisation of $\partial_H(S\|K\|R\|L)$ illustrated

A.2 The Concurrent Alternating Bit Protocol

A.2.1 Module P

The picture in Figure A.2 is an illustration to the equations in Section 5.4.2. I left out labels of internal steps for readability.

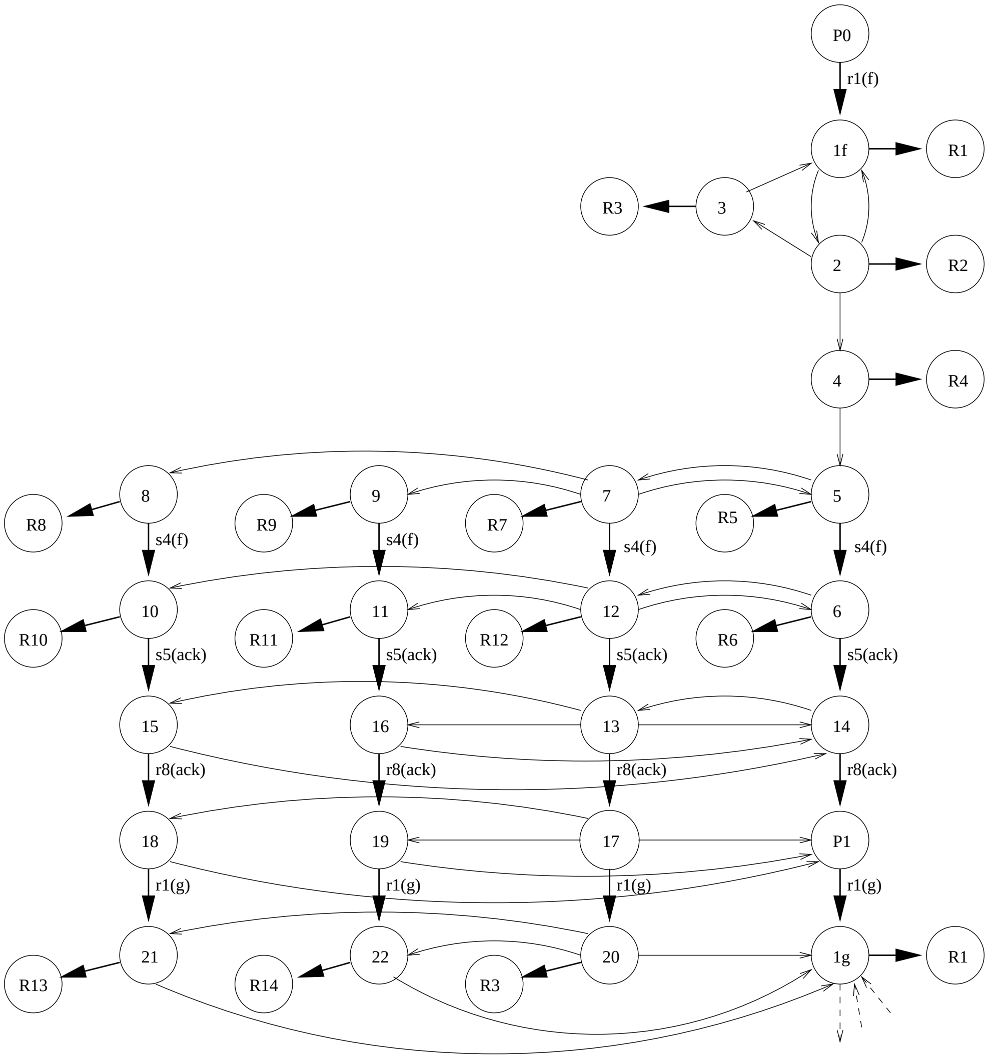


Figure A.2: Linearisation of P_n

Figure A.3 is the same as Figure A.2 except that redundant states have been omitted.

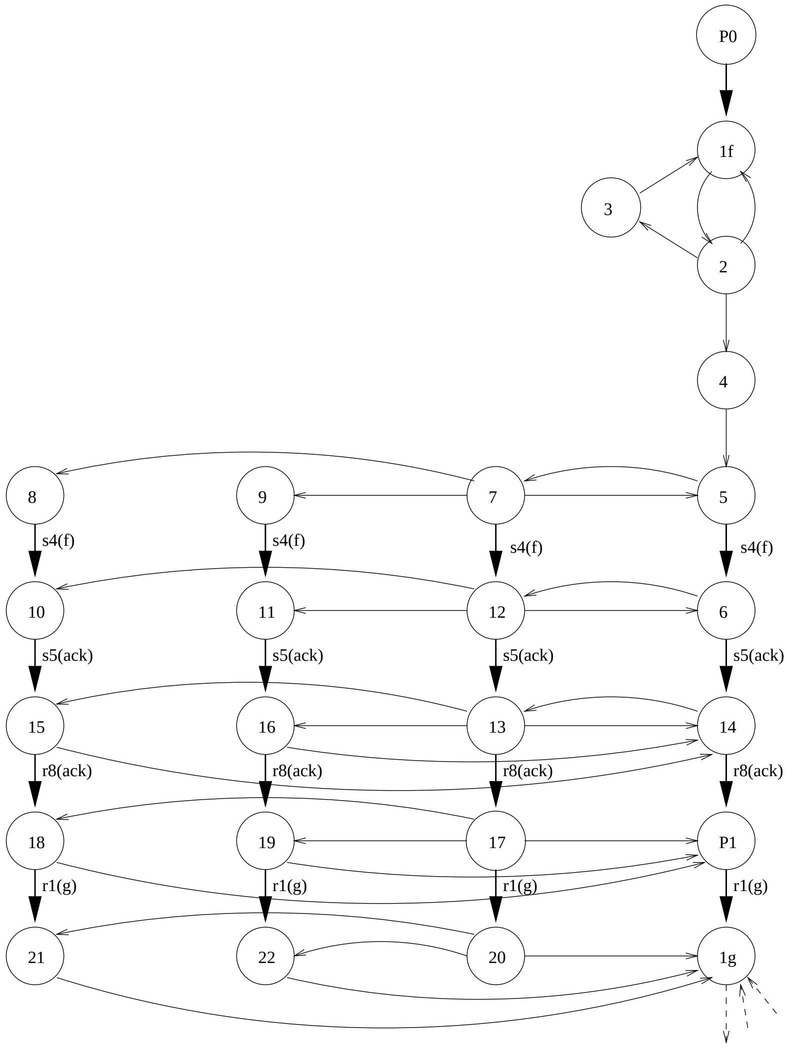


Figure A.3: Illustration of the FLS of $\tilde{P}_n$ from Section 5.4.3.

Figure A.4 illustrates the FLS from Section 5.4.4 derived after abstraction. Now unlabeled transitions are ι-steps.

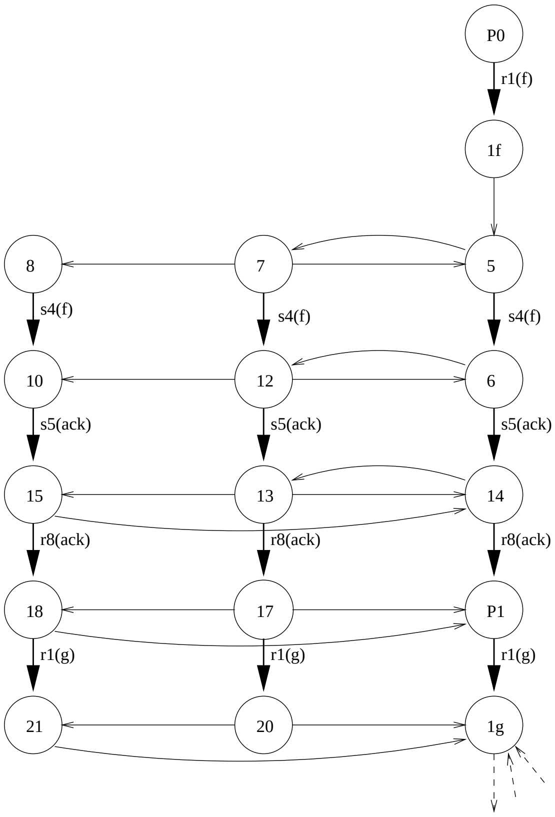


Figure A.4: Illustration of the FLS of $\iota_{I_P}(\tilde{P}_n)$

A.2.2 Module Q

The picture in Figure A.5 is an illustration to the equations in Section 5.5.2. I left out labels of internal steps for readability.

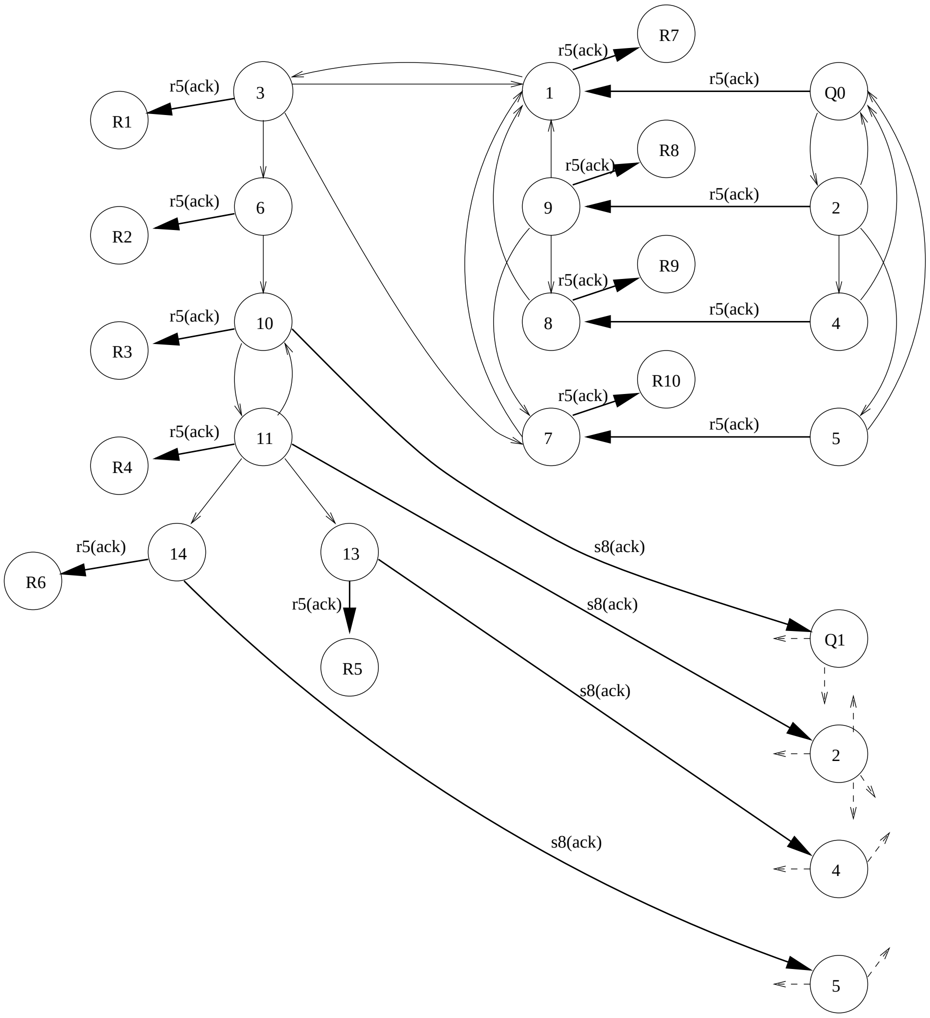


Figure A.5: Linearisation of Q_n

Figure A.6 is the same as Figure A.5 except that redundant states have been omitted.

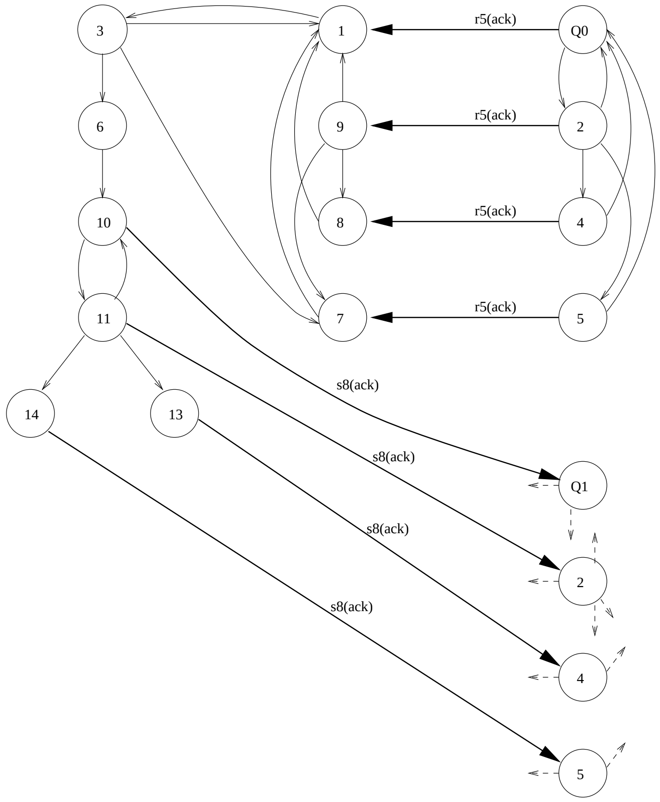


Figure A.6: Illustration of the FLS of $\tilde{Q}_n$ from Section 5.5.3.

Figure A.7 illustrates the FLS from Section 5.5.4 derived after abstraction. Now unlabeled transitions are ι-steps.

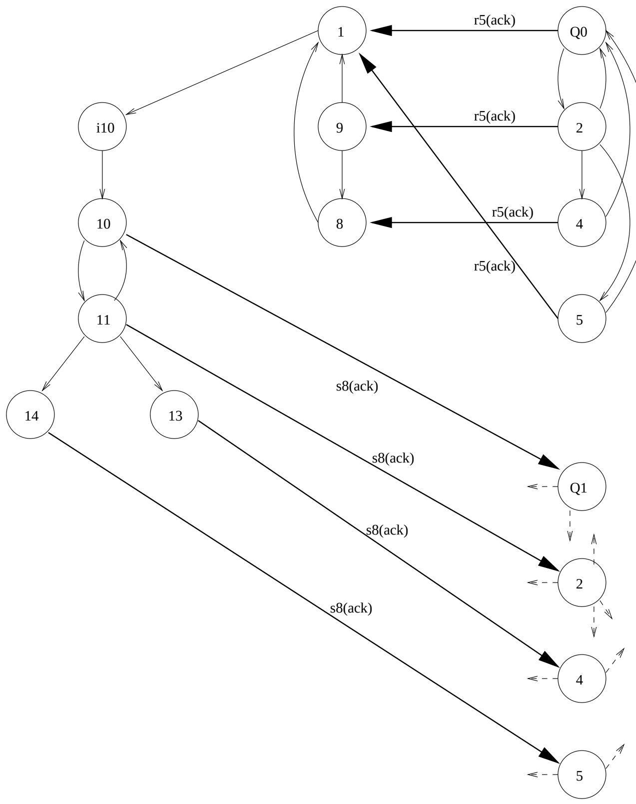


Figure A.7: Illustration of the FLS of $\iota_{I_Q}(\tilde{Q}_n)$ from Section 5.5.4.

A.3 The PAR Protocol

Figure A.8 illustrates the linearisation of $\theta \circ \partial_H(S^t \| K \| R \| L)$.

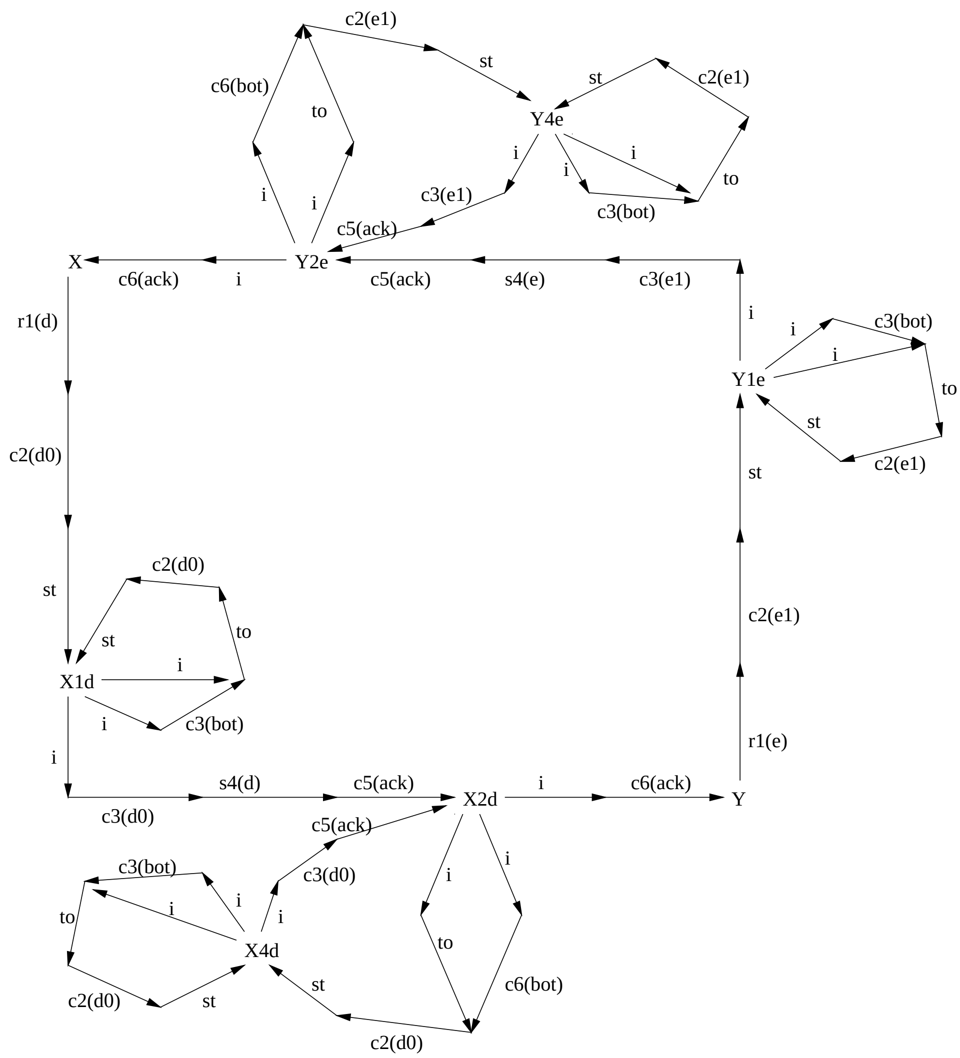


Figure A.8: Illustration of the linearisation of $\theta \circ \partial_H(S^t \| K \| R \| L)$

Technical Reports of the Programming Research Group

Note: These reports can be obtained using the technical reports overview on our WWW site (`http://www.wins.uva.nl/research/prog/reports/`) or by anonymous ftp to `ftp.wins.uva.nl`, directory `pub/programming-research/reports/`.

[P9806] M. B. van der Zwaag. *Some Verifications in Process Algebra with Iota.*

[P9805] A. Sellink and C. Verhoef. *Development, Assessment, and Reengineering of Language Descriptions.*

[P9804] A. Sellink and C. Verhoef. *Native Patterns.*

[P9803] J.J. Brunekreef. *Annotated Algebraic Specification of the Syntax and Semantics of the Programming Language* Alma-0.

[P9802] W. Fokkink and C. Verhoef. *Conservative Extension in Positive/Negative Conditional Term Rewriting with Applications to Software Renovation Factories.*

[P9801] P. Klint and C. Verhoef. *Evolutionary Software Engineering: A Component-based Approach.*

[P9726] J.A. Bergstra and A. Ponse. *Grid Protocol Specifications.*

[P9725] J.A. Bergstra and A. Ponse. *Process Algebra Primitives for File Transfer.*

[P9724] J.A. Bergstra and A. Ponse. *Process Algebra with Four-Valued Logic.*

[P9723] J.A. Bergstra and A. Ponse. *Kleene's Three-Valued Logic and Process Algebra.*

[P9722] J.A. Bergstra and A. Ponse. *Bochvar-McCarthy Logic and Process Algebra.*

[P9721] M.P.A. Sellink and C. Verhoef. *Reflections on the Evolution of COBOL.*

[P9720] M. van der Graaf. *A Specification of Box to HTML in ASF+SDF.*

[P9719] M. van den Brand, M.P.A. Sellink, and C. Verhoef. *Current Parsing Techniques in Software Renovation Considered Harmful.*

[P9718] M. de Jonge. *Reuse of ASF+SDF Specifications by means of Renaming.*

[P9717] A. Sellink (Ed.). *Participants Proceedings of the 2nd International Workshop on the Theory and Practice of Algebraic Specifications.*

[P9717] E. Visser. *From Context-free Grammars with Priorities to Character Class Grammars.*

[P9716] H.M. Sneed. *Dealing with the Dual Crisis — Year 2000 and Euro — What Reverse Engineering can do to Help.*

[P9715] W. Fokkink and C. Verhoef. *An SOS Message: Conservative Extension in Higher Order Positive/Negative Conditional Term Rewriting.*

[P9714] M. van den Brand, M.P.A. Sellink, and C. Verhoef. *Control Flow Normalization for COBOL/CICS Legacy Systems.*

[P9713] B. Diertens. *Simulation and Animation of Process Algebra Specifications.*

[P9712] V. Partington. *Implementation of an Imperative Programming Language with Backtracking.*

[P9711] L. Moonen. *A Generic Architecture for Data Flow Analysis to Support Reverse Engineering.*

[P9710] B. Luttik and E. Visser. *Specification ofRewriting Strategies.*

[P9709] J.A. Bergstra and M.P.A. Sellink. *An Arithmetical Module for Rationals and Reals.*

[P9708] E. Visser. *Character Classes.*

[P9707] E. Visser. *Scannerless Generalized-LR Parsing.*

[P9706] E. Visser. *A Family of Syntax Definition Formalisms.*

[P9705] M.G.J. van den Brand, M.P.A. Sellink, and C. Verhoef. *Generation of Components for Software Renovation Factories from Context-free Grammars.*

[P9704] P.A. Olivier. *Debugging Distributed Applications Using a Coordination Architecture.*

[P9703] H.P. Korver and M.P.A. Sellink. *A Formal Axiomatization for Alphabet Reasoning with Parametrized Processes.*

[P9702] M.G.J. van den Brand, M.P.A. Sellink, and C. Verhoef. *Reengineering COBOL Software Implies Specification of the Underlying Dialects.*

[P9701] E. Visser. *Polymorphic Syntax Definition.*

[P9618] M.G.J. van den Brand, P. Klint, and C. verhoef. *Re-engineering needs Generic Programming Language Technology.*

[P9617] P.I. Manuel. *ANSI Cobol III in SDF + an ASF Definition of a Y2K Tool.*

[P9616] P.H. Rodenburg. *A Complete System of Four-valued Logic.*

[P9615] S.P. Luttik and P.H. Rodenburg. *Transformations of Reduction Systems.*

[P9614] M.G.J. van den Brand, P. Klint, and C. Verhoef. *Core Technologies for System Renovation.*

[P9613] L. Moonen. *Data Flow Analysis for Reverse Engineering.*

[P9612] J.A. Hillebrand. *Transforming an ASF+SDF Specification into a ToolBus Application.*

[P9611] M.P.A. Sellink. *On the conservativity of Leibniz Equality.*

[P9610] T.B. Dinesh and S.M. Üsküdarlı. *Specifying input and output of visual languages.*

[P9609] T.B. Dinesh and S.M. Üsküdarlı. *The VAS formalism in VASE.*

[P9608] J.A. Hillebrand. *A small language for the specification of Grid Protocols.*