Cylindric Process Algebras with Conditionals give Substitutionless pCRL

S.P. Luttik

Information Systems (INS)

SEN-R9912 May 1999

Report SEN-R9912
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

Cylindric Process Algebras with Conditionals give Substitutionless pCRL

S.P. Luttik
Bas.Luttik@cwi.nl
CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Programming Research Group, University of Amsterdam,
Kruislaan 403, NL-1098 SJ Amsterdam, The Netherlands

ABSTRACT
Theories that are designed to reason about processes and the information they exchange usually also include a construction to quantify over this information. We mention the input prefix mechanism of, e.g., the π-calculus and the operation $\sum$ for alternative quantification over data of μCRL. These constructions are implemented as binders, and hence imply a relatively complicated notion of substitution that takes the binding of these operations into account. As a consequence, the defining axioms of such constructions are not suitable for purely equational reasoning, or, to put it differently, proof systems for process theories that contain binders are not algebraic.

In this paper, we shall define the variety of cylindric process algebras with conditionals, and we prove that, for data with built-in equality and built-in Skolem functions, it is an algebraic semantics of a subsystem of μCRL. Thus we obtain a proof system for a class of process algebras with data and alternative quantification from which the complicated notion of substitution is eliminated.

1991 Mathematics Subject Classification: 03G15; 08A70; 68Q70
1991 Computing Reviews Classification System: F.1.1; F.3.3; F.4.1
Keywords and Phrases: Process Algebra, Cylindric Algebra, Algebraic Logic, Alternative Quantification, Input Prefixing.
Note: Research supported by the Netherlands Organization for Scientific Research (NWO) under contract SION 612-33-008. Work carried out under project SEN 2.1 Process Specification and Analysis.

1. Introduction

Motivated by remarks of Milner (1983), Bergstra and Klop (1984) put forward a purely algebraic approach to concurrency theory. They defined a class of varieties of algebras with operations for alternative ($+$), sequential ($\cdot$) and parallel ($\|$) composition by means of a parametrised set ACP of equational axioms. To reason formally about the interaction of processes with the data that they communicate, Groote and Ponse (1994b) introduced the specification formalism μCRL, an extension of ACP. It features actions of the form $a(\bar{t})$ that are parametrised with a sequence of data expressions $\bar{t}$, a conditional construct $p \lhd b \rhd q$ that chooses between two alternatives p and q depending on some boolean expression b over data, and an operation $\sum$ that we would like to call *alternative quantification over data*.

This latter operation has our particular interest. It is defined in μCRL as a binder. For instance, if v is a variable that ranges over some data type $D = \{d_0, d_1, d_2, \dots\}$, then $\sum_v p(v)$ stands for the (possibly infinite) summation

$$p(d_0) + p(d_1) + p(d_2) + \cdots .$$

The main use of alternative quantification is to model the action that *inputs* an arbitrary data element. For instance, the term $\sum_v \mathtt{read}(v) \cdot p(v)$ refers to the process that inputs an arbitrary element of D and proceeds as the particular instantiation of $p(v)$ with this element.

In the style of ACP, Groote and Ponse (1994a) gave equational axioms also for the binder $\sum$. For instance, one of the axioms that appears in their paper, is

(SUM1) $\sum_v x \approx x$.

Obviously, one should not be able to conclude from this axiom that $\sum_v p = p$, for all processes p, so this axiom is not sound in combination with equational logic as such. An instance of this axiom with a process p for the variable x is correct if, and only if, the variable v does not occur freely in p.

Groote and Ponse (1994a) account for this by a redefinition of substitution such that free variables never become bound by substitution and bound variables are never substituted for. Their proof system resembles the system of natural deduction for classical predicate logic with identity, but without rules for universal and existential quantification.

Groote and Luttik (1998) presented a proof system for pCRL, a subsystem of μCRL that includes alternative quantification, that stays closer to equational logic. It is based on *generalised equational logic*, as they call it. This is equational logic, extended with a congruence rule for binders and an appropriately redefined notion of substitution. The authors prove that their system is complete for strong bisimulation provided that the data has built-in equality and Skolem functions. Completeness must be understood in the following sense: the equation $p \approx q$ (with p and q possibly containing free data variables) is provable in the system if, and only if, the processes denoted by p and q are strongly bisimilar for every instantiation of the variables with data elements.

It seems a rather severe restriction that the data must have built-in Skolem functions; it means that the *first-order* theory of the data has quantifier elimination, and hence is decidable. However, another result in the same paper is that strong bisimulation with arbitrary computable data is at least Π^0_4-hard, which provides strong evidence that a more general completeness theorem is not easy to achieve.

Reasoning by purely equational means is considered simple and intuitive, but in a setting with binders it becomes relatively complicated. The notion of a variable occurring free in a term, and the notion of substitution that takes bound variables into account are notoriously hard to define and to work with. Tarski (1965) was able to eliminate these notions from the formalisation of first-order predicate logic with identity by observing that if the variable v does not occur (at all) in t, then the formula

$$\varphi[v := t] \leftrightarrow \exists v(v \approx t \wedge \varphi)$$

is a tautology. Intuitively, the notion of substitution is already implicit in first-order logic.

With respect to pCRL we can make a similar observation if the data has built-in equality and Skolem functions: if v is a variable that does not occur in a data term t, then

$$p[v := t] \approx \sum_v p \lhd \mathrm{eq}(v, t) \rhd \delta.$$

This underlies the present paper. We shall employ it to find an *algebraic semantics* (see Blok and Pigozzi (1989)) for pCRL: a variety of algebras that satisfy the identities of pCRL such that the algebra of pCRL-terms modulo derivability is a member of this variety.

An algebraic semantics of first-order logic is the variety of cylindric algebras (see Henkin *et al.* (1971, 1985)). These are boolean algebras that are extended with operations that reflect existential quantification and equality. Existential quantification is represented in cylindric algebras by an infinite supply of unary operations c_κ, one for each variable v_κ in the enumeration of all variables $v_0, v_1, v_2, \ldots$; these operations are called *cylindrifications*. The expression $c_\kappa\varphi$ refer to the formula $\exists v_\kappa\varphi$. Equality is represented by means of an infinite supply of constants $d_{\kappa\lambda}$; these constants are called *diagonal elements*. The constant $d_{\kappa\lambda}$ corresponds to the atomic formula $v_\kappa \approx v_\lambda$.

We shall introduce in this paper the variety of cylindric process algebras. Roughly, these are two-sorted algebras consisting of a cylindric algebra and a process algebra extended with operations for

alternative quantification and conditionals. The cylindric algebra replaces the boolean algebra of $p\mathsf{CRL}$. We justify this by the assumption that the data has built-in equality and built-in Skolem functions; diagonal elements and cylindrifications are definable in the boolean algebra if this is the case. Alternative quantification is represented by means of an infinite supply of unary operations s_κ that we call *cylindric summations*; the expression $s_\kappa p$ corresponds to the process $\sum_{v_\kappa} p$.

This paper is organised as follows. In §2 we shall define cylindric algebras and explain their connection with first-order predicate logic; in §3 we explain how data specifications with built-in equality and Skolem functions can be interpreted as cylindric algebras. In §4 we introduce the proof system $p\mathsf{CRL}$ of Groote and Luttik (1998), and we show that without losing expressive or demonstrative power we can restrict the set of $p\mathsf{CRL}$-terms. In §5 we introduce the variety of cylindric process algebras with conditionals, and we prove that the algebra of $p\mathsf{CRL}$-terms is isomorphic with a cylindric process algebra with conditionals that is free in a particular subvariety of cylindric process algebras with conditionals.

2. Cylindric Algebras

We assume that the reader has some basic knowledge of universal algebra and equational logic. For thorough treatments of these subject we refer to McKenzie *et al.* (1987) and Burris and Sankappanavar (1981), and for the generalisation to the many-sorted case to Meinke and Tucker (1992). We keep our notation consistent with McKenzie *et al.* (1987). In particular, if K is a variety of similar algebras, then we use $\mathfrak{F}_\mathsf{K}(X)$ to denote a free algebra in K with free generating set X (such an algebra is unique up to isomorphism).

Referring to Birkhoff's HSP Theorem, we make no distinction between equational specifications and the varieties that they induce. Suppose that S is an equational specification, Then we write $\mathfrak{F}_\mathsf{S}(X)$ to denote a free algebra in the variety induced by S with free generating set X. If Σ is the signature of S, then we obtain, using Birkhoff's Completeness Theorem for Equational Logic, a concrete example of such a free algebra by constructing the algebra $\mathfrak{T}_\Sigma(X)$ of Σ-terms over X and taking the quotient $\mathfrak{T}_\Sigma(X)/EqTh(\mathsf{S})$ of $\mathfrak{T}_\Sigma(X)$ over the equational theory $EqTh(\mathsf{S})$ of S, which is a congruence on $\mathfrak{T}_\Sigma(X)$. In the sequel, we shall often treat $\mathfrak{F}_\mathsf{S}(X)$ as if it were constructed in this way, and view its elements as equivalence classes of terms; if t is a Σ-term over X, then we denote the element of $\mathfrak{F}_\mathsf{S}(X)$ that contains t by $\overline{t}$.

A *cylindric algebra of dimension* ω (CA_ω) is an algebraic structure

$$\mathfrak{C} = \langle \mathrm{C}, \vee, \wedge, \neg, \bot, \top, c_\kappa, d_{\kappa\lambda}\rangle_{\kappa,\lambda<\omega}$$

such that for every $x, y \in C$ and $\kappa, \lambda, \mu < \omega$

(CA0)	the structure $\langle C, \vee, \wedge, \neg, \bot, \top\rangle$ is a boolean algebra (see Table 1);	
(CA1)	$c_\kappa \bot$	$\approx \bot$;
(CA2)	$x \vee c_\kappa x$	$\approx c_\kappa x$;
(CA3)	$c_\kappa(x \wedge c_\kappa y)$	$\approx c_\kappa x \wedge c_\kappa y$;
(CA4)	$c_\kappa c_\lambda x$	$\approx c_\lambda c_\kappa x$;
(CA5)	$d_{\kappa\kappa}$	$\approx \top$;
(CA6)	if $\kappa \neq \lambda, \mu$, then $d_{\lambda\mu} \approx c_\kappa(d_{\lambda\kappa} \wedge d_{\kappa\mu})$;	
(CA7)	if $\kappa \neq \lambda$, then $c_\kappa(d_{\kappa\lambda} \wedge x) \wedge c_\kappa(d_{\kappa\lambda} \wedge \neg x) \approx \bot$.	

The operations c_κ are called *cylindrifications* and the constants $d_{\kappa\lambda}$ are called *diagonal elements*.

The theory of cylindric algebras was introduced by Tarski and his students to give a purely algebraic treatment of first-order predicate logic. In the remainder of this section we shall discuss those parts of this connection that are of relevance to the present paper; for a detailed account we refer to Henkin *et al.* (1985).

(BA1)	$x \wedge y \approx y \wedge x$	$x \vee y \approx y \vee x$
(BA2)	$x \wedge (y \wedge z) \approx (x \wedge y) \wedge z$	$x \vee (y \vee z) \approx (x \vee y) \vee z$
(BA3)	$x \wedge x \approx x$	$x \vee x \approx x$
(BA4)	$x \wedge (x \vee y) \approx x$	$x \vee (x \wedge y) \approx x$
(BA5)	$(x \wedge y) \vee (x \wedge z) \approx x \wedge (y \vee z)$	$(x \vee y) \wedge (x \vee z) \approx x \vee (y \wedge z)$
(BA6)	$x \wedge \bot \approx \bot$	$x \vee \top \approx \top$
(BA7)	$x \wedge \neg x \approx \bot$	$x \vee \neg x \approx \top$

Table 1: The axioms for boolean algebras.

We fix a countably infinite sequence $\mathcal{V} = \langle v_\xi \mid \xi < \omega \rangle$ of *variables*. Let Λ be a first-order language without operation symbols; we assume that Λ is given by a pair $\langle \mathcal{R}, \rho \rangle$, where $\mathcal{R}$ is a set of *relation symbols* and ρ is a function from $\mathcal{R}$ to ω that associates with every $r \in \mathcal{R}$ an *arity*. We denote by F_Λ the set of first-order Λ-formulae in variables from $\mathcal{V}$. If $\Gamma \subseteq F_\Lambda$ and $\varphi \in F_\Lambda$, then we write $\Gamma \vdash \varphi$ if φ is derivable from Γ in some proof system that is complete for first-order predicate logic with identity (see e.g. Shoenfield (1967)). Consider the algebraic structure

$$\mathfrak{Fm}^\Lambda = \langle F_\Lambda, \vee, \wedge, \neg, \bot, \top, \exists v_\kappa, v_\kappa \approx v_\lambda \rangle_{\kappa,\lambda<\omega}$$

that is naturally induced on the set of first-order Λ-formulae by the logical connectives. This algebra is similar to CA_ω's via the interpretation of $(\exists v_\kappa)$ as the unary operation c_κ, and $v_\kappa \approx v_\lambda$ as the constant $d_{\kappa\lambda}$. For each $\Gamma \subseteq F_\Lambda$, the relation

$$\equiv_\Gamma = \{\langle \varphi, \psi \rangle \mid \Gamma \vdash \varphi \leftrightarrow \psi\}$$

is a congruence on this algebra, and the quotient is a cylindric algebra of dimension ω (cf. Henkin *et al.* (1985)). We shall refer to it by $\mathfrak{Fm}^\Lambda_\Gamma$, i.e.,

$$\mathfrak{Fm}^\Lambda_\Gamma = \mathfrak{Fm}^\Lambda / \equiv_\Gamma.$$

The key observation now is that if u and v are distinct variables, and $\varphi[u := v]$ is the first-order Λ-formula that is obtained by substituting v for u in φ, then the formula

$$\varphi[u := v] \leftrightarrow (\exists u)(u \approx v \wedge \varphi)$$

is a tautology. We can use this to associate with every first-order Λ-formula φ a logically equivalent *restricted* formula φ^r; this is a formula that is constructed with the boolean connectives and quantifiers from atomic formulae $v_\kappa \approx v_\lambda$ and $r(v_0, \ldots, v_{\rho(r)-1})$, with $\kappa, \lambda < \omega$ and $r \in \mathcal{R}$. For instance, if r is a relation symbol of arity 2, and u and w are variables different from the first two elements v_0 and v_1 of $\mathcal{V}$, then the first-order formula

$$r(u, w) \leftrightarrow \exists v_0 \exists v_1 (v_0 \approx u \wedge v_1 \approx w \wedge r(v_0, v_1))$$

is a tautology, and the righthand side of the bi-implication is restricted.

Since, in a restricted formula, each relation symbol r is always followed by the fixed list of variables $v_0, \ldots, v_{\rho(r)-1}$, the mapping g defined by

$$\begin{array}{ll} g(r(v_0, \ldots, v_{\rho(r)-1})) = r & g(\neg\varphi) = \neg g(\varphi) \\ g(v_\kappa \approx v_\lambda) = d_{\kappa\lambda} & g(\varphi \vee \psi) = g(\varphi) \vee g(\psi) \\ g(\exists v_\kappa \varphi) = c_\kappa g(\varphi) & g(\varphi \wedge \psi) = g(\varphi) \wedge g(\psi) \end{array}$$

gives a one-to-one correspondence between the restricted formulae and the CA_ω-terms over $\mathcal{R}$. We extend g to a mapping from the set of all first-order Λ-formulae by defining $g(\varphi) = g(\varphi^r)$.

Let $\mathfrak{F}_{\mathsf{CA}_\omega}(\mathcal{R})$ be the cylindric algebra that is freely generated by $\mathcal{R}$. We define $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ as the quotient of $\mathfrak{F}_{\mathsf{CA}_\omega}(\mathcal{R})$ over the congruence that is induced by the identities

$$(\text{CA8}) \quad c_\kappa r \approx r, \qquad \text{for all } r \in \mathcal{R} \text{ and } \rho(r) \leq \kappa < \omega.$$

Clearly, this algebra is isomorphic to the quotient algebra formed by the algebra of CA_ω-terms over $\mathcal{R}$ and the congruence generated by CA0–CA8. Hence, we may view the elements of $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ as equivalence classes of terms; we denote the element of $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ that contains the term t by $\overline{t}$. Although variables do not occur in CA_ω-terms over $\mathcal{R}$, the statement $v_\kappa \notin FV(\varphi)$ has a cylindric counterpart.

Proposition 2.1 If $v_\kappa \notin FV(\varphi)$, then CA_ω+CA8 $\vdash c_\kappa g(\varphi) \approx g(\varphi)$.

Proof. By induction on the length of φ^r. □

By a *filter* of a CA_ω $\mathfrak{C}$ we mean a subset $\mathcal{F}$ of the universe of $\mathfrak{C}$ such that

1. $\top \in \mathcal{F}$;
2. if $x, \neg x \vee y \in \mathcal{F}$, then $y \in \mathcal{F}$; and
3. if $x \in \mathcal{F}$ and $\kappa < \omega$, then $\neg c_\kappa \neg x \in \mathcal{F}$.

Every filter of $\mathfrak{C}$ determines a congruence of $\mathfrak{C}$ and we denote the quotient of $\mathfrak{C}$ over this congruence by $\mathfrak{C}/\mathcal{F}$. Moreover, there is a one to one correspondence between the filters of $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ and the first-order Λ-theories. For $\Gamma \subseteq F_\Lambda$, let $\mathcal{F}_\Gamma$ be the filter of $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ that is generated by the set

$$\{\overline{g(\varphi)} \mid \varphi \in \Gamma\}$$

(i.e., the smallest filter in which this set is contained). We then have that

Theorem 2.2 $\mathfrak{Fm}^\Lambda_\Gamma$ and $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}/\mathcal{F}_\Gamma$ are isomorphic via the mapping $\varphi/{\equiv_\Gamma} \mapsto \overline{g(\varphi)}$.

Proof. See Theorems 4.3.6, 4.3.25 and 4.3.27 of Henkin *et al.* (1985). □

So far, we have restricted our discussion to first-order languages without operation symbols. Now, suppose that Λ is a first-order language *with* operation symbols, and let Γ be a set of first-order Λ-formulae; one can show that $\mathfrak{Fm}^\Lambda_\Gamma$ is a cylindric algebra. We shall now describe how to associate with Λ and Γ a free cylindric algebra and a filter such that $\mathfrak{Fm}^\Lambda_\Gamma$ is isomorphic to the quotient (cf. also Monk (1965)).

Let Λ' be the first-order language without operation symbols that consists of the relation symbols of Λ, and additionally contains for every n-ary operation symbol f in Λ an n+1-ary relation symbol r_f. We inductively associate with every Λ-term t and every variable u in $\mathcal{V}$ a Λ'-formula $\varphi_{u,t}$ as follows:

1. if t is a variable, then $\varphi_{u,t}$ is the atomic formula $u \approx t$; and
2. if t is of the form $f(t_0, \ldots, t_{n-1})$, where f is an n-ary operation symbol and the $t_0, \ldots, t_{n-1}$ are terms, then $\varphi_{u,t}$ is the formula

$$(\forall u_0) \cdots (\forall u_{n-1})(\varphi_{u_0,t_0} \wedge \cdots \wedge \varphi_{u_{n-1},t_{n-1}} \to r_f(u_0, \ldots, u_{n-1}, u)),$$

where $u_0, \ldots, u_{n-1}$ is the initial segment of the list of variables in $\mathcal{V}$ that do not occur in t or u.

We can now define a mapping h from atomic Λ-formulae to atomic Λ'-formulae by

$$t_0 \approx t_1 \mapsto (\forall u_0)(\forall u_1)(\varphi_{u_0,t_0} \wedge \varphi_{u_1,t_1} \to u_0 \approx u_1)$$
$$r(t_0, \ldots, t_{n-1}) \mapsto (\forall u_0) \cdots (\forall u_{n-1})(\varphi_{u_0,t_0} \wedge \ldots \varphi_{u_{n-1},t_{n-1}} \to r(u_0, \ldots, u_{n-1}));$$

and extend it to a mapping from Λ-formulae to Λ'-formulae in the usual way. Notice that h is defined in such a way that

Proposition 2.3 $FV(\varphi) = FV(h(\varphi))$.

We denote by Ω_Λ the least set of Λ'-formulae that contains for every n-ary operation symbol f in Λ the formulae

$$(\forall x_0)\dots(\forall x_{n-1})(\exists x_n) r_f(x_0,\dots,x_n);\text{ and}$$
$$(\forall x_0)\dots(\forall x_{n-1})(\forall y)(\forall z)(r_f(x_0,\dots,x_{n-1},y) \wedge r_f(x_1,\dots,x_{n-1},z) \to y \approx z).$$

Then there exists for every Λ'-formula ψ a Λ-formula φ such that $\Omega_\Lambda \vdash \psi \leftrightarrow h(\varphi)$, and $\Gamma \vdash \varphi$ if, and only if, $h(\Gamma) \cup \Omega_\Lambda \vdash h(\varphi)$; hence $\mathfrak{Fm}^\Lambda_\Gamma \cong \mathfrak{Fm}^{\Lambda'}_{h(\Gamma)\cup\Omega_\Lambda}$.

We define $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ as the quotient of $\mathfrak{F}^{\Lambda'}_{\mathsf{CA}_\omega}$ over the congruence that is induced by the identities

$$\begin{array}{lll} (\text{CA}9) & \neg c_0 \cdots c_{n-1} \neg c_n r_f & \approx \top, \text{ and} \\ (\text{CA}10) & \neg c_0 \cdots c_{n+1} \neg (r_f \wedge \neg d_{nn+1} \wedge c_n(r_f \wedge d_{nn+1})) & \approx \top, \end{array}$$

for every n-ary operation symbol f in Λ. By Theorem 2.2 the algebras $\mathfrak{Fm}^{\Lambda'}_{\Omega_\Lambda}$ and $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ are isomorphic. Let $\mathcal{F}_\Gamma$ be the filter of $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}$ that is generated by the set

$$\{\overline{g(h(\varphi))} \mid \varphi \in \Gamma\}.$$

By Theorems 4.3.6, 4.3.25, and 4.3.27 of Henkin *et al.* (1985), the algebras $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}/\mathcal{F}_\Gamma$ and $\mathfrak{Fm}^{\Lambda'}_{h(\Gamma)\cup\Omega_\Lambda}$ are isomorphic, and we have the following

Theorem 2.4 $\mathfrak{Fm}^\Lambda_\Gamma$ and $\mathfrak{F}^\Lambda_{\mathsf{CA}_\omega}/\mathcal{F}_\Gamma$ are isomorphic via the mapping $\varphi/{\equiv_\Gamma} \mapsto \overline{g(h(\varphi))}$.

3. Data specifications

In the theory $p\mathsf{CRL}$, the data must be given by means of a many-sorted equational specification $\mathsf{D} = \langle \Delta, \mathrm{E}\rangle$ that contains the axioms of boolean algebra (see Table 1); such an equational specification is called a *data specification*. For easier explanation of our ideas we shall focuss on a certain kind of data specifications that have a nice correspondence with theories of first-order predicate logic.[1]

We assume that our data specifications have a two-sorted signature; they contain a sort $\mathbb{b}$ for *booleans* and a sort $\mathbb{d}$ of *data elements*. Apart from the connectives $\vee$, $\wedge$, $\neg$, $\bot$ and $\top$ of boolean algebras, the function declarations that involve the sort $\mathbb{b}$ have the form

$$r\colon \mathbb{d}^n \to \mathbb{b} \qquad \text{for } n \in \omega;$$

the remaining function declarations are of the form

$$f\colon \mathbb{d}^n \to \mathbb{d} \qquad \text{for } n \in \omega.$$

We shall occasionally refer to the function declarations of the first kind as *relations*, and to those of the second kind as *functions*. Let us fix a data specification $\mathsf{D} = \langle \Delta, \mathrm{E}\rangle$ of this kind; we denote by $\mathrm{E}_\mathbb{b}$ the set of axioms of sort $\mathbb{b}$ (it contains the axioms in Table 1), and by $\mathrm{E}_\mathbb{d}$ the set of axioms of sort $\mathbb{d}$.

We may regard Δ as the first-order language that has the relations of Δ as relation symbols and the functions of Δ as operation symbols. We associate with E a set $\mathrm{E}_{\mathsf{fol}}$ of first-order Δ-formula in the following way:

$$\mathrm{E}_{\mathsf{fol}} = \{b_1 \leftrightarrow b_2 \mid (b_1 \approx b_2) \in \mathrm{E}_\mathbb{b}\} \cup \mathrm{E}_\mathbb{d}.$$

[1] We think that it is quite straightforward to generalise our approach to arbitrary data specifications, although this requires a connection between many-sorted first-order logic and cylindric algebras.

Let $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$ denote the free algebra that is generated by the set $\mathcal{V}$ of variables of sort $\mathfrak{d}$. The element $\overline{b}$ of $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$ that contains the boolean term b is a subset of the element $b/{\equiv_{\mathrm{E_{fol}}}}$ of $\mathfrak{Fm}^{\Delta}_{\mathrm{E_{fol}}}$. Also, if b_1 and b_2 are first-order Δ-formulae not containing quantifiers and $\mathrm{E_{fol}} \vdash b_1 \leftrightarrow b_2$, then there exists a derivation of $b_1 \leftrightarrow b_2$ without applications of axioms and rules that involve quantifiers. Hence, the mapping defined by

$$\overline{b} \mapsto b/{\equiv_{\mathrm{E_{fol}}}} \tag{3.1}$$

is an embedding of the boolean reduct of $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$ into the boolean part of $\mathfrak{Fm}^{\Delta}_{\mathrm{E_{fol}}}$.

We shall now find sufficient conditions on D that allow us to define diagonal elements and cylindrifications in $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$. Diagonal elements are definable if there exists a predicate that is logically equivalent to the identity predicate of first-order logic.

Definition 3.1 We say that D has *built-in equality* if there exists a boolean Δ-term $\mathrm{eq}(u, v)$ in variables u and v of sort $\mathfrak{d}$ such that for all Δ-terms t_1 and t_2 of sort $\mathfrak{d}$

$$\mathrm{E_{fol}} \vdash \mathrm{eq}(t_1, t_2) \leftrightarrow t_1 \approx t_2.$$

To define cylindrifications, we use a notion from model theory (cf. Chang and Keisler (1990)).

Definition 3.2 Let φ be a first-order Δ-formula and let v be a variable of sort $\mathfrak{d}$. We call a Δ-term $t^{\varphi}_{v}(u_1, \ldots, u_n)$ of sort $\mathfrak{d}$ a *Skolem function* for φ and v if, and only if,

$$\mathrm{E_{fol}} \vdash (\exists v)\varphi(v, u_1, \ldots, u_n) \rightarrow \varphi(t^{\varphi}_{v}(u_1, \ldots, u_n), u_1, \ldots, u_n).$$

If there exists a Skolem function t^{b}_{v} for every boolean term b and for every variable v of sort $\mathfrak{d}$, then we say that D has built-in Skolem functions.

If D has both built-in equality and built-in Skolem functions, then we define, for $\kappa, \lambda < \omega$,

$$d_{\kappa\lambda} = \mathrm{eq}(v_\kappa, v_\lambda);$$
$$c_\kappa b = b[v_\kappa := t^{b}_{v_\kappa}], \text{ where } t^{b}_{v_\kappa} \text{ is the Skolem function for } b \text{ and } v_\kappa.$$

Adding these operations to the boolean reduct of $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$ yields an algebraic structure that is similar to cylindric algebras; we shall refer to it as the *cylindric part of* $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$.

Theorem 3.3 If D has built-in equality and built-in Skolem functions, then the cylindric part of $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$ is isomorphic with $\mathfrak{Fm}^{\Delta}_{\mathrm{E_{fol}}}$ via the mapping $\overline{b} \mapsto b/{\equiv_{\mathrm{E_{fol}}}}$.

Proof. If $t^{\varphi}_{v}(u_1, \ldots, u_n)$ is the Skolem function for φ and v, then

$$\mathrm{E_{fol}} \vdash (\exists v)\varphi(v, u_1, \ldots, u_n) \leftrightarrow \varphi(t(u_1, \ldots, u_n), u_1, \ldots, u_n)$$

(the implication from left to right is from Definition 3.2, and the other implication is a tautology), so every first-order Δ-formula is logically equivalent to one without quantifiers. Also, since D has built-in equality we can replace every atomic formula of the form $t_1 \approx t_2$ by the logically equivalent boolean Δ-term $\mathrm{eq}(t_1, t_2)$. Hence, for every first-order Δ-formula φ there exists a boolean Δ-term b such that $\mathrm{E_{fol}} \vdash \varphi \leftrightarrow b$; so the mapping defined in (3.1) is the required isomorphism. □

With every first-order Λ-theory Γ we can associate a filter $\mathcal{F}_\Gamma$ of the cylindric algebra $\mathfrak{F}^{\Lambda}_{\mathsf{CA}_\omega}$ such that $\mathfrak{Fm}^{\Lambda}_{\Gamma} \cong \mathfrak{F}^{\Lambda}_{\mathsf{CA}_\omega}/\mathcal{F}_\Gamma$. We define $\mathcal{F}_{\mathsf{D}}$ as the filter associated to $\mathrm{E_{fol}}$ and combine Theorems 2.2 and 3.3 into the following result

Corollary 3.4 If D has built-in equality and built-in Skolem functions, then the cylindric part of $\mathfrak{F}_{\mathsf{D}}(\mathcal{V})$ is isomorphic with $\mathfrak{F}^{\Delta}_{\mathsf{CA}_\omega}/\mathcal{F}_{\mathsf{D}}$ via the mapping $\overline{b} \mapsto \overline{g(h(b))}$.

(A1)	$x + y$	$\approx y + x$	(A6)	$x + \delta$	$\approx x$
(A2)	$x + (y + z)$	$\approx (x + y) + z$	(A7)	$\delta \cdot x$	$\approx \delta$
(A3)	$x + x$	$\approx x$			
(A4)	$(x + y) \cdot z$	$\approx x \cdot z + y \cdot z$			
(A5)	$x \cdot (y \cdot z)$	$\approx (x \cdot y) \cdot z$			

Table 2: The axioms for process algebras with deadlock.

4. The deductive system pCRL

We shall now define the deductive system pCRL that was presented in Groote and Luttik (1998) to reason about processes with data. We show that one can make similar observations about pCRL as we made about first-order predicate logic in §2: without decreasing expressive and demonstrative power we can restrict the set of terms of our language.

Let us fix a two-sorted data specification $\mathsf{D} = \langle \Delta, \mathrm{E} \rangle$ with built-in equality and built-in Skolem functions, and a sort symbol $\mathbb{p}$ that refers to a set of processes. An n-ary *action declaration* for Δ is a function declaration of the form

$$\mathsf{a} \colon \mathbb{d}^n \to \mathbb{p}.$$

We fix a nonempty set A of action declarations for Δ. Let Π be the signature that is the extension of Δ with the sort $\mathbb{p}$, the action declarations in A, and

1. a constant δ of sort $\mathbb{p}$;
2. binary function declarations $(\_ + \_) \colon \mathbb{p}\mathbb{p} \to \mathbb{p}$ and $(\_ \cdot \_) \colon \mathbb{p}\mathbb{p} \to \mathbb{p}$;
3. a ternary function declaration $(\_ \lhd \_ \rhd \_) \colon \mathbb{p}\mathbb{b}\mathbb{p} \to \mathbb{p}$; and
4. a binder declaration $\sum : \mathbb{p}$.

We shall call Π a pCRL-*signature*. The constant δ and the binary operations $+$ and $\cdot$ are from the theory ACP (Bergstra and Klop (1984)); $+$ stands for alternative composition, and $\cdot$ stands for sequential composition. In Table 2 we have listed the axioms of ACP that concern these operations. The conditional $(\_ \lhd \_ \rhd \_)$ (read $p \lhd c \rhd q$ as "if c then p else q") and the binder $\sum$ to quantify over data stem from μCRL (Groote and Ponse (1994b)), an extension of ACP that allows reasoning about the combination of processes and data. Terms over Π will be considered modulo α-conversion.

In Table 3 we have listed the axioms for the conditional and alternative quantification over data. The schemes Sum3 and Sum4 define an axiom for every instantiation of p and q with Π-terms of sort $\mathbb{p}$ in variables from $\mathcal{V}$. The scheme Sum1 defines an axiom for every instantiation of p with a pCRL-term in which the variable v does not occur freely; similar remarks can be made about the schemes Sum5 and Sum12. The scheme Ae defines an axiom for every action declaration in A; if a is an n-ary action declaration, then $\bar{u}$ refers to a list of n variables $u_0, \ldots, u_{n-1}$, and similarly for $\bar{w}$; by $\mathrm{eq}(\bar{u}, \bar{w})$ we have abbreviated the boolean term $\mathrm{eq}(u_0, w_0) \wedge \cdots \wedge \mathrm{eq}(u_{n-1}, w_{n-1})$.

We would like to reason equationally with these axioms. However, the presence of alternative quantification complicates matters, since equational logic has no facility for binders. We need to incorporate a congruence rule for each binder in the signature; in our particular setting this rule takes the form

$$\frac{p \approx q}{\sum_v p \approx \sum_v q}.$$

(Cond1)	$x \lhd \top \rhd y$	$\approx x$	
(Cond2)	$x \lhd \bot \rhd y$	$\approx y$	
(Cond3)	$x \lhd b \rhd y$	$\approx x \lhd b \rhd \delta + y \lhd \neg b \rhd \delta$	
(Cond4)	$(x \lhd b_1 \rhd \delta) \lhd b_2 \rhd \delta$	$\approx x \lhd b_1 \wedge b_2 \rhd \delta$	
(Cond5)	$(x \lhd b_1 \rhd \delta) + (x \lhd b_2 \rhd \delta)$	$\approx x \lhd b_1 \vee b_2 \rhd \delta$	
(Cond6)	$(x \lhd b \rhd \delta)y$	$\approx xy \lhd b \rhd \delta$	
(Cond7)	$(x + y) \lhd b \rhd \delta$	$\approx x \lhd b \rhd \delta + y \lhd b \rhd \delta$	
(Sca)	$(x \lhd b \rhd \delta)(y \lhd b \rhd \delta)$	$\approx (xy \lhd b \rhd \delta)$	

(Ae) $\mathtt{a}(\bar{u}) \lhd \mathrm{eq}(\bar{u}, \bar{w}) \rhd \delta \approx \mathtt{a}(\bar{w}) \lhd \mathrm{eq}(\bar{u}, \bar{w}) \rhd \delta$

(Sum1)	$\sum_v p$	$\approx p$	if $v \notin FV(p)$
(Sum3)	$\sum_v p$	$\approx \sum_v p + p$	
(Sum4)	$\sum_v (p + q)$	$\approx \sum_v p + \sum_v q$	
(Sum5)	$(\sum_v p)q$	$\approx \sum_v pq$	if $v \notin FV(q)$
(Sum12)	$(\sum_v p) \lhd c \rhd \delta$	$\approx \sum_v p \lhd c \rhd \delta$	if $v \notin FV(c)$

Table 3: The axioms of pCRL for conditionals and alternative quantification over data.

Moreover, we need to redefine substitution in such a way that free variables do not become bound by substitution, and bound variables are never substituted for. The resulting system is called *generalised* equational logic by Groote and Luttik (1998).

Let t_1 and t_2 be Π-terms of the same sort. We write $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash t_1 \approx t_2$ if the equation $t_1 \approx t_2$ is derivable from the axioms of D, those in Table 2, and those in Table 3, by means of generalised equational logic. We give an example of a proof in $p\mathsf{CRL}(\mathsf{D}, \mathrm{A})$.

Proposition 4.1 $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash \sum_u \sum_v p \approx \sum_v \sum_u p$.

Proof. We have the following derivation:

$$\begin{aligned} \textstyle\sum_u \sum_v p &\approx \textstyle\sum_v \sum_u \sum_u \sum_v p && \text{(by Sum1)} \\ &\approx \textstyle\sum_v \sum_u \sum_u \sum_v p + \sum_v \sum_u p && \text{(by Sum3 and Sum4)} \\ &\approx \textstyle\sum_u \sum_v p + \sum_v \sum_u p && \text{(by Sum1);} \end{aligned}$$

so $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash \sum_u \sum_v p \approx \sum_u \sum_v p + \sum_v \sum_u p \approx \sum_v \sum_u p$. □

Let $T_{\mathbb{b}}[\mathcal{V}]$ and $T_{\mathbb{p}}[\mathcal{V}]$ be the sets of Π-terms of sorts $\mathbb{b}$ and $\mathbb{p}$, and consider the algebra[2]

$$\langle T_{\mathbb{b}}[\mathcal{V}], T_{\mathbb{p}}[\mathcal{V}], \vee, \wedge, \neg, \bot, \top, +, \cdot, \delta, (\_ \lhd \_ \rhd \delta), \textstyle\sum_{v_\kappa} \rangle_{\kappa < \omega}$$

The relation $\{\langle t_1, t_2 \rangle \mid p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash t_1 \approx t_2 \ \&\ t_1, t_2 \in T_{\mathbb{b}}[\mathcal{V}] \text{ or } t_1, t_2 \in T_{\mathbb{p}}[\mathcal{V}]\}$ is a congruence on this algebra. Since D has built-in equality and built-in Skolem functions, we can define diagonal elements and cylindrifications in the quotient; we refer to the resulting algebra as $\mathfrak{Pcrl}(\mathsf{D}, \mathrm{A})$. The fact that D has built-in Skolem functions also has another important consequence: if $\sum_v$ only binds variables in the boolean argument of a conditional, then it can be eliminated.

Proposition 4.2 If $v \notin FV(p)$, and t_v^b is a Skolem function for b and v, then

$$p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash \textstyle\sum_v p \lhd b \rhd \delta \approx p \lhd b[v := t_v^b] \rhd \delta.$$

[2]For reasons that will become clear later on, we define on this algebra a binary conditional $(\_ \lhd \_ \rhd \delta)$ instead of the ternary $(\_ \lhd \_ \rhd \_)$.

PROOF. Since t_v^b is a Skolem function for b and v,

$$\mathsf{D} \vdash b[v := t_v^b] \approx b[v := t_v^b] \vee b; \tag{*}$$

so we can make the following derivation

$$\begin{aligned} p \lhd b[v := t_v^b] \rhd \delta &\approx \textstyle\sum_v p \lhd b[v := t_v^b] \vee b \rhd \delta && \text{(by SUM1 and *)} \\ &\approx \textstyle\sum_v (p \lhd b[v := t_v^b] \rhd \delta + p \lhd b \rhd \delta) && \text{(by COND5)} \\ &\approx \textstyle\sum_v p \lhd b[v := t_v^b] \rhd \delta + \sum_v p \lhd b \rhd \delta && \text{(by SUM4)} \\ &\approx p \lhd b[v := t_v^b] \rhd \delta + \textstyle\sum_v p \lhd b \rhd \delta && \text{(by SUM1).} \end{aligned}$$

On the other hand, the identity $\sum_v p \lhd b \rhd \delta \approx \sum_v p \lhd b \rhd \delta + p \lhd b[v := t_v^b] \rhd \delta$ is an instance of SUM3, so with A1

$$\textstyle\sum_v p \lhd b \rhd \delta \approx \sum_v p \lhd b \rhd \delta + p \lhd b[v := t_v^b] \rhd \delta \approx p \lhd b[v := t_v^b] \rhd \delta.$$ □

Substitution is definable in $p\mathsf{CRL}$.

PROPOSITION 4.3 If D has built-in equality and Skolem functions, and v and w are distinct variables, then $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash p[v := w] \approx \sum_v p \lhd \mathrm{eq}(v, w) \rhd \delta$, for all process terms p.

PROOF. If v and w are distinct variables, then $v \notin FV(p[v := w])$, and w is a Skolem function for $\mathrm{eq}(v, w)$ and v, since $\mathrm{eq}(w, w) \approx \top$ by Theorem 3.3, so

$$\begin{aligned} p[v := w] &\approx p[v := w] \lhd \mathrm{eq}(w, w) \rhd \delta && \text{(by COND1)} \\ &\approx \textstyle\sum_v p[v := w] \lhd \mathrm{eq}(v, w) \rhd \delta && \text{(by Proposition 4.2).} \end{aligned}$$

It remains to prove that $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash p \lhd \mathrm{eq}(v, w) \rhd \delta \approx p[v := w] \lhd \mathrm{eq}(v, w) \rhd \delta$. We proceed by induction on the structure of p. If p is δ, then $p = p[v := w]$. If $\bar{t}$ is a sequence of data terms, then $\mathrm{eq}(v, w) \approx \mathrm{eq}(v, w) \wedge \mathrm{eq}(\bar{t}[v := w], \bar{t})$ by Theorem 3.3; so if p is of the form $\mathsf{a}(\bar{t})$, then we have the following derivation

$$\begin{aligned} \mathsf{a}(\bar{t}) \lhd \mathrm{eq}(v, w) \rhd \delta &\approx \mathsf{a}(\bar{t}) \lhd \mathrm{eq}(v, w) \wedge \mathrm{eq}(\bar{t}[v := w], \bar{t}) \rhd \delta \\ &\approx \mathsf{a}(\bar{t})[v := w] \lhd \mathrm{eq}(v, w) \wedge \mathrm{eq}(\bar{t}[v := w], \bar{t}) \rhd \delta && \text{(by AE)} \\ &\approx \mathsf{a}(\bar{t})[v := w] \lhd \mathrm{eq}(v, w) \rhd \delta. \end{aligned}$$

The cases where p is of the form $p_1 + p_2$, $p_1 \cdot p_2$, or $p_1 \lhd b \rhd p_2$ follow from the induction hypothesis and COND7, SCA, COND3 and COND4. If p is of the form $\sum_u p'$, then by α-conversion we may assume that $u \neq v, w$, so the result follows from the induction hypothesis and SUM12. □

COROLLARY 4.4 Suppose that D has built-in equality and built-in Skolem functions. If $v \notin FV(p)$, then

$$p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash \textstyle\sum_v (\sum_w p \lhd \mathrm{eq}(v, w) \rhd \delta) \lhd \mathrm{eq}(v, w) \rhd \delta \approx p.$$

PROOF. If $v = w$, then the corollary follows from COND1 and SUM1; if $v \neq w$, then the corollary follows from Proposition 4.3. □

We call a $p\mathsf{CRL}$-term of sort $\mathbb{p}$ *restricted* if every n-ary action declaration that occurs in it is immediately followed by the fixed initial segment $v_0, \ldots, v_{n-1}$ of $\mathcal{V}$ and each occurrence of the conditional has δ as its rightmost argument.

LEMMA 4.5 If D has built-in equality and built-in Skolem functions, then for every $p\mathsf{CRL}$-term p of sort $\mathbb{p}$ there exists a restricted $p\mathsf{CRL}$-term p^r such that $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash p \approx p^r$.

PROOF. By axiom COND3 we can replace every occurrence of a conditional $p \lhd c \rhd q$ with $q \neq \delta$ by its equivalent $p \lhd c \rhd \delta + q \lhd \neg c \rhd \delta$. Then it remains to prove the lemma for terms of the form $\mathsf{a}(\bar{t})$, with a an n-ary action declaration, and $\bar{t}$ a sequence of terms of the appropriate length that is not the initial segment $\bar{v} = v_0, \ldots, v_{n-1}$ of $\mathcal{V}$.
Let $\bar{w} = w_0, \ldots, w_{n-1}$ be the initial segment of the list that is obtained by removing from $\mathcal{V}$ the variables that occur in $\bar{v}$ and in $\bar{t}$. We abbreviate by $\sum_{\bar{v}}$ the sequence $\sum_{v_0} \cdots \sum_{v_{n-1}}$ and by $\sum_{\bar{w}}$ the sequence $\sum_{w_0} \cdots \sum_{w_{n-1}}$. With Proposition 4.1 and axiom SUM12 we get from Corollary 4.4 that

$$p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash \textstyle\sum_{\bar{w}}(\sum_{\bar{v}} \mathsf{a}(\bar{v}) \lhd \mathrm{eq}(\bar{v}, \bar{w}) \rhd \delta) \lhd \mathrm{eq}(\bar{v}, \bar{w}) \rhd \delta \approx \mathsf{a}(\bar{v});$$

so we obtain $\sum_{\bar{w}}(\sum_{\bar{v}} \mathsf{a}(\bar{v}) \lhd \mathrm{eq}(\bar{v}, \bar{w}) \rhd \delta) \lhd \mathrm{eq}(\bar{t}, \bar{w}) \rhd \delta \approx \mathsf{a}(\bar{t})$ as a substitution instance, and the lefthand side of this equation is restricted. □

If we consider only the restricted terms of sort $\mathbb{p}$, then we can replace the axioms COND1 and COND2 by

$$\begin{array}{ll} (\mathrm{COND1}^r) & x \lhd \top \rhd \delta \approx x \quad \text{and} \\ (\mathrm{COND2}^r) & x \lhd \bot \rhd \delta \approx \delta. \end{array}$$

Let us define a proof system $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A})$ on the set of restricted pCRL-terms by replacing COND1 and COND2 by $\mathrm{COND1}^r$ and $\mathrm{COND2}^r$, deleting COND3 and the instances of AE, and adding as axioms the restricted instances of Corollary 4.4. The previous lemma showed that $p\mathsf{CRL}(\mathsf{D}, \mathrm{A})$ and $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A})$ have the same expressive power; we now show that they also have the same demonstrative power.

LEMMA 4.6 If $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash p \approx q$, and p' and q' are restricted pCRL-terms such that $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash p \approx p', q \approx q'$, then $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A}) \vdash p' \approx q'$.

PROOF. We associate with every pCRL-term p of sort $\mathbb{p}$ the particular restricted pCRL-term p^r of the proof of Lemma 4.5; this association is surjective, for if p is restricted, then $p = p^r$. Hence, it suffices to show that if $p\mathsf{CRL}(\mathsf{D}, \mathrm{A}) \vdash p \approx q$, then $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A}) \vdash p^r \approx q^r$.
We shall first deal with the special case that $p \approx q$ is a substitution instance of an axiom of $p\mathsf{CRL}(\mathsf{D}, \mathrm{A})$. If it is an instance of COND1, then we derive $p^r \lhd \top \rhd \delta + q^r \lhd \bot \rhd \delta \approx p^r + \delta \approx p^r$ by $\mathrm{COND1}^r$, $\mathrm{COND2}^r$ and A6. For instances of COND2 we have a similar derivation in $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A})$. The only nontrivial instances of COND3 are those with $y = \delta$; we derive

$$\begin{array}{ll} p^r \lhd b \rhd \delta + \delta \lhd \neg b \rhd \delta & \\ \quad \approx p^r \lhd b \rhd \delta + \delta \lhd \neg b \rhd \delta + \delta \lhd \top \rhd \delta & \text{(by A6 and COND1)} \\ \quad \approx p^r \lhd b \rhd \delta + \delta \lhd \top \rhd \delta & \text{(by COND5)} \\ \quad \approx p^r \lhd b \rhd \delta & \text{(by COND1 and A6).} \end{array}$$

If $p \approx q$ is an instance of AE, then we find $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A}) \vdash p^r \approx q^r$ as an instance of Corollary 4.4, or with axioms SUM12 and COND4. If $p \approx q$ is an instance of some other axiom of $p\mathsf{CRL}(\mathsf{D}, \mathrm{A})$, then the verification that $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A}) \vdash p^r \approx q^r$ is straightforward.
The general case now follows by induction on a shortest derivation of $p \approx q$. □

We denote by $T_{\mathbb{p}}[\mathcal{V}]^r$ the set of restricted Π-terms of sort $\mathbb{p}$, and consider the algebra

$$\langle T_{\mathbb{b}}[\mathcal{V}], T_{\mathbb{p}}[\mathcal{V}]^r, \vee, \wedge, \neg, \bot, \top, +, \cdot, \delta, (\_ \lhd \_ \rhd \delta), \textstyle\sum_{v_\kappa} \rangle_{\kappa < \omega}$$

The deductive system $p\mathsf{CRL}^r(\mathsf{D}, \mathrm{A})$ induces a congruence on this algebra, and cylindrifications and diagonal elements are definable in the quotient; we refer to the resulting algebra as $\mathfrak{Pcrl}^r(\mathsf{D}, \mathrm{A})$. The following result is an immediate consequence of Lemmas 4.5 and 4.6.

THEOREM 4.7 $\mathfrak{Pcrl}(\mathsf{D}, \mathrm{A}) \cong \mathfrak{Pcrl}^r(\mathsf{D}, \mathrm{A})$.

5. Cylindric Process Algebras with Conditionals

In the previous section we introduced the deductive system $p\mathsf{CRL}(\mathsf{D},\mathrm{A})$ to reason about processes with data. However, this deductive system is not algebraic: it is not a pure extension of equational logic. Namely, the presence of the binder $\sum$ required that we redefined the notion of substitution. We shall now show that $p\mathsf{CRL}(\mathsf{D},\mathrm{A})$ is algebraizable in the sense of Blok and Pigozzi (1989): there exists a variety of algebras that satisfy the identities that are derivable in $p\mathsf{CRL}(\mathsf{D},\mathrm{A})$ and the algebra $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$ is a member of this variety. Such a variety is called an *algebraic semantics* of $p\mathsf{CRL}(\mathsf{D},\mathrm{A})$.

Let us first consider the extension of process algebras with unary operations s_κ.

Definition 5.1 A *cylindric process algebra of dimension* ω (CPA_ω) is a structure

$$\langle \mathrm{P}, +, \cdot, \delta, s_\kappa\rangle_{\kappa<\omega}$$

that satisfies for every $x, y \in \mathrm{P}$ and $\kappa, \lambda < \omega$

(Cs0)	the structure $\langle \mathrm{P}, +, \cdot, \delta\rangle$ is a process algebra with deadlock (see Table 2);	
(Cs1)	$s_\kappa s_\lambda x$	$\approx s_\lambda s_\kappa x$
(Cs2)	$s_\kappa s_\kappa x$	$\approx s_\kappa x$
(Cs3)	$x + s_\kappa x$	$\approx s_\kappa x$
(Cs4)	$s_\kappa(x+y)$	$\approx s_\kappa x + s_\kappa y$
(Cs5)	$s_\kappa(x \cdot s_\kappa y)$	$\approx s_\kappa x \cdot s_\kappa y$
(Cs6)	$s_\kappa \delta$	$\approx \delta$

The operations s_κ we call *cylindric summations*.

If we interpret $\sum_{v_\kappa}$ as the unary operation s_κ, then the $\mathfrak{p}$-reduct of $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$ is an algebra similar to CPA_ω's.

Theorem 5.2 The $\mathfrak{p}$-reduct of $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$ is a cylindric process algebra of dimension ω.

Proof. It is clear that it is a process algebra, and the axioms Cs3 and Cs4 hold by Sum3 and Sum12. From Proposition 4.1 we know that Cs1 holds in the $\mathfrak{p}$-reduct of $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$, and since $v_\kappa \notin FV(\sum_{v_\kappa} p)$, the validity of axioms Cs2 and Cs5 follow from Sum1 and Sum5. Finally, since $v_\kappa \notin FV(\delta)$, the validity of Cs6 follows from Sum1. □

Definition 5.3 A CPA_ω *with conditionals* is a two-sorted algebra that consists of

a CA_ω $\langle \mathrm{C}, \vee, \wedge, \neg, \bot, \top, c_\kappa, d_{\kappa\lambda}\rangle_{\kappa,\lambda<\omega}$,
a CPA_ω $\langle \mathrm{P}, +, \cdot, \delta, s_\kappa\rangle_{\kappa<\omega}$, and
an operation $\_ :\!\rightarrow \_ : \mathrm{C} \times \mathrm{P} \to \mathrm{P}$

such that for every $b, c \in \mathrm{C}$, $x, y \in \mathrm{P}$, and $\kappa < \omega$

(Gc1)	$\top :\!\rightarrow x$	$\approx x$
(Gc2)	$\bot :\!\rightarrow x$	$\approx \delta$
(Gc3)	$b :\!\rightarrow (x+y)$	$\approx b :\!\rightarrow x + b :\!\rightarrow y$
(Gc4)	$(b \vee c) :\!\rightarrow x$	$\approx b :\!\rightarrow x + c :\!\rightarrow x$
(Gc5)	$b :\!\rightarrow (c :\!\rightarrow x)$	$\approx (b \wedge c) :\!\rightarrow x$
(Gc6)	$b :\!\rightarrow (x \cdot y)$	$\approx (b :\!\rightarrow x) \cdot y$
(Gc7)	$b :\!\rightarrow (x \cdot y)$	$\approx (b :\!\rightarrow x) \cdot (b :\!\rightarrow y)$
(Cs7)	$s_\kappa(b :\!\rightarrow s_\kappa x)$	$\approx c_\kappa b :\!\rightarrow s_\kappa x$
(Cs8)	$s_\kappa(c_\kappa b :\!\rightarrow x)$	$\approx c_\kappa b :\!\rightarrow s_\kappa x$
(Cs9)	$s_\kappa\, d_{\kappa\lambda} :\!\rightarrow s_\lambda\, d_{\kappa\lambda} :\!\rightarrow s_\kappa x$	$\approx x$

The axioms Gc1–6 originate from Baeten and Bergstra (1992), but we adopt the numbering of Bergstra *et al.* (1994a).

If we interpret $(\_ \lhd \_ \rhd \delta)$ as the binary operation $\_ :\!\rightarrow \_$, then the algebra $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$ is similar to CPA_ω's.

THEOREM 5.4 If D has built-in equality and Skolem functions, then $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$ is a CPA_ω with conditionals.

PROOF. In view of Theorem 5.2 it suffices to show that the axioms GC1–GC7 and CS7–CS9 hold in $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$. The axioms GC1 and GC2 are instances of COND1 and COND2, and the axioms GC3–7 correspond to COND7, COND5, COND4, COND6 and SCA (in that order). Since $v_\kappa \notin FV(\sum_{v_\kappa} p)$, the validity of CS7 and CS9 follows from Proposition 4.2 and Corollary 4.4, and if $t^b_{v_\kappa}$ is the Skolem function for b and v_κ, then $v_\kappa \notin FV(b[x := t^b_{v_\kappa}])$, so the validity of CS8 is by axiom SUM12. □

The cylindric counterpart of a first-order language is a set of relation symbols $\mathcal{R}$ together with a set of axioms (generated by the scheme CA8) that specifies the arity of these symbols. We shall now define the cylindric counterpart of a set of action declarations. Let $\mathcal{A}$ be a set of *action symbols* and let ρ be a mapping into the set ω that associates with every $a \in \mathcal{A}$ an arity $\rho(a)$. We specify the arity of each action symbol by means of the identities

$$\text{(CS10)}\quad s_\kappa a \approx a, \qquad \text{for all } \rho(a) \le \kappa < \omega.$$

Let $\Sigma = \langle \mathcal{R}, \mathcal{A}, \rho\rangle$, where $\mathcal{R}$ and $\mathcal{A}$ are disjoint sets of relation symbols and action symbols, and ρ is a mapping from $\mathcal{R} \cup \mathcal{A}$ to ω that assigns to every relation symbol and every action symbol an arity. Now, consider the free CPA_ω with conditionals $\mathfrak{F}_{\mathsf{CPA}_\omega}(\mathcal{R},\mathcal{A})$ that is generated by the sets $\mathcal{R}$ and $\mathcal{A}$ (obviously, we use $\mathcal{R}$ to generate the cylindric algebra and $\mathcal{A}$ to generate the process algebra). We define the algebra $\mathfrak{F}^\Sigma_{\mathsf{CPA}_\omega}$ as the quotient of $\mathfrak{F}_{\mathsf{CPA}_\omega}(\mathcal{R},\mathcal{A})$ over the congruence induced on it by the schemes CA8 and CS10. We regard the elements of $\mathfrak{F}^\Sigma_{\mathsf{CPA}_\omega}$ as equivalence classes of CPA_ω-terms over $\mathcal{R}$ and $\mathcal{A}$.

Obviously, we can associate with every $p\mathsf{CRL}$-signature Π a triple $\Sigma = \langle \mathcal{R}, \mathcal{A}, \rho\rangle$ such that the data signature Δ and the set of action declarations A on which Π is based can be reconstructed from the pairs $\langle \mathcal{R}, \rho\rangle$ and $\langle \mathcal{A}, \rho\rangle$; we shall write $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}$ to denote $\mathfrak{F}^\Sigma_{\mathsf{CPA}_\omega}$. Notice that $\mathfrak{F}^\Delta_{\mathsf{CA}_\omega}$ is the cylindric algebra of $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}$, and recall that we associated with every data specification D with built-in equality and Skolem functions a filter $\mathcal{F}_\mathsf{D}$ of the algebra $\mathfrak{F}^\Delta_{\mathsf{CA}_\omega}$. We define $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$ as the quotient of $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}$ over the smallest congruence that contains the congruence generated by $\mathcal{F}_\mathsf{D}$ on $\mathfrak{F}^\Delta_{\mathsf{CA}_\omega}$. It is, of course, our aim to show that $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$ is isomorphic with $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$.

Let g and h be the mappings defined in §2, and define $f = g \circ h$; f maps each boolean Δ-term to a CA_ω-term over $\mathcal{R}$. We extend f to a mapping from restricted $p\mathsf{CRL}$-terms to CPA_ω-terms over $\mathcal{R}$ and $\mathcal{A}$ as follows:

$$\begin{array}{ll} f(\mathsf{a}(v_0,\ldots,v_{\rho(r)-1})) = \mathsf{a} & \text{for all } \mathsf{a} \in \mathrm{A} \\ f(p \lhd c \rhd \delta) = f(c) :\!\rightarrow f(p) & f(p \cdot q) = f(p) \cdot f(q) \\ f(p+q) = f(p) + f(q) & f(\textstyle\sum_{v_\kappa}) p = s_\kappa f(p) \end{array}$$

We shall prove that the mapping f gives rise to an isomorphism from $\mathfrak{Pcrl}^r(\mathsf{D},\mathrm{A})$ to $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$. We need the following analogue of Proposition 2.1.

PROPOSITION 5.5 If $v_\kappa \notin FV(p)$, then CPA_ω+CA8+CS10 $\vdash s_\kappa f(p) \approx f(p)$.

PROOF. By induction on the length of p, using Propositions 2.1 and 2.3. □

THEOREM 5.6 $\mathfrak{Pcrl}^r(\mathsf{D},\mathrm{A})$ and $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$ are isomorphic.

PROOF. By Corollary 3.4, the data parts of $\mathfrak{Pcrl}^r(\mathsf{D},\mathrm{A})$ and $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$ are isomorphic via the mapping $\overline{b} \mapsto \overline{f(b)}$, so it remains to show that f also gives rise to an isomorphism from the process part of $\mathfrak{Pcrl}^r(\mathsf{D},\mathrm{A})$ to the process part of $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$ that respects the operation $\_ :\!\rightarrow \_$.
It is immediate from SUM1 that CS10 holds in $\mathfrak{Pcrl}^r(\mathsf{D},\mathrm{A})$, so with Theorem 5.4 we conclude that if $\overline{f(p)} = \overline{f(q)}$, then $p\mathsf{CRL}^r(\mathsf{D},\mathrm{A}) \vdash p \approx q$. We shall now establish that the converse also holds by proving that the axioms of $p\mathsf{CRL}^r(\mathsf{D},\mathrm{A})$ hold in $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$.

The axioms COND1^r, COND2^r, COND4–COND7 and SCA correspond to the axioms GC1, GC2, GC5, GC4, GC6, GC3 and GC7 (in that order). The validity of axiom SUM1 follows from Proposition 5.5, and the axioms SUM3 and SUM4 correspond to CS3 and CS4. For SUM5 we have the following derivation

$$\begin{aligned} s_\kappa f(p) \cdot f(q) &\approx s_\kappa f(p) \cdot s_\kappa f(q) && \text{(Proposition 5.5)} \\ &\approx s_\kappa(f(p) \cdot s_\kappa f(q)) && \text{(CS5)} \\ &\approx s_\kappa(f(p) \cdot f(q)) && \text{(Proposition 5.5).} \end{aligned}$$

The instances of Corollary 4.4 are derivable in a similar fashion with CS9, and SUM12 can be derived with CS5 and Propositions 2.1 and 2.3.
Hence, $p\mathsf{CRL}^r(\mathsf{D},\mathrm{A}) \vdash p \approx q$ if, and only if, $\overline{f(p)} = \overline{f(q)}$, and since f is a bijection between restricted $p\mathsf{CRL}$-terms and CPA_ω-terms over $\mathcal{R}$ and A, the association $\overline{p} \mapsto \overline{f(p)}$ is an isomorphism between the process parts of $\mathfrak{Pcrl}^r(\mathsf{D},\mathrm{A})$ and $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$. □

COROLLARY 5.7 The algebras $\mathfrak{Pcrl}(\mathsf{D},\mathrm{A})$ and $\mathfrak{F}^\Pi_{\mathsf{CPA}_\omega}/\mathcal{F}_\mathsf{D}$ are isomorphic.

PROOF. By Theorem 4.7. □

6. Conclusions

Usually, theories that are designed to reason formally about processes and the information that they exchange include a construction to quantify over this information. For instance, the π-calculus of Milner *et al.* (1992) and the message-passing process algebras of Hennessy (1991) involve *input* prefixes; these are action prefixes '$\mathsf{a}(v)$;' that bind the variable v in their argument; $\mathsf{a}(v);p$ is the process that inputs an arbirary datum d along channel a and continue as the particular instantiation of p with d substituted for v.

The presence of binders in such theories implies an awkward notion of substitution. Bergstra *et al.* (1994b) show that this can be circumvented by importing the equational theory of combinatory logic, using the fundamental combinators I, K and S to express substitution equationally. In this paper we proved that in the case of $p\mathsf{CRL}$ it is not necessary to include another calculus to deal with substitution, provided that the data has built-in equality and Skolem functions. We did this by finding an algebraic semantics for $p\mathsf{CRL}$ in the form of cylindric process algebras with conditionals. Having such an algebraic semantics enables us to reason about processes and data in a purely equational way, and we now have the theory of universal algebra at our disposal to find new results about process algebras with data.

Our algebraic semantics of $p\mathsf{CRL}$ is based on the observation that we can use the process term $\sum_v p \lhd \mathrm{eq}(v,t) \rhd \delta$ to denote the result of substituting the data term t for all free occurrences of the variable v in process term p. Notice that we make use of a special feature of $p\mathsf{CRL}$: the conditional restricts the set of data over which $\sum_v$ quantifies to only those elements that are equal to t. This feature cannot be expressed if quantification is combined with action prefixing. Hence, our method does not directly apply to languages with input prefixing, such as the π-calculus. In this respect, we mention the fusion calculus of Parrow and Victor (1998), an extension of the π-calculus in which action prefixing is separated from quantification. We conjecture that it can be provided with an algebraic semantics in a similar fashion.

Acknowledgements The author thanks Vincent van Oostrom and Piet Rodenburg for carefully reading earlier versions of this paper, and Jan Friso Groote and Mark van der Zwaag for comments.

References

Baeten, J. C. M. and Bergstra, J. A. (1992). Process algebra with signals and conditions. In M. Broy, editor, *Programming and Mathematical Method*, volume 88 of *NATO ASI Series F: Computer and System Sciences*, pages 273–323. Springer-Verlag. Proceedings of the 1990 International Summer School in Marktoberdorf.

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

Bergstra, J. A., Ponse, A., and van Wamel, J. J. (1994a). Process algebra with backtracking. In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, *Proceedings of the REX Symposium "A Decade of Concurrency: Reflections and Perspectives"*, volume 803 of *Lecture Notes in Computer Science*, pages 46–91. Springer-Verlag.

Bergstra, J. A., Bethke, I., and Ponse, A. (1994b). Process algebra with combinators. In E. Börger, Y. Gurevich, and K. Meinke, editors, *Computer science logic (Swansea, 1993)*, volume 832 of *Lecture Notes in Computer Science*, pages 36–65. Springer-Verlag.

Blok, W. J. and Pigozzi, D. (1989). Algebraizable logics. *Mem. Amer. Math. Soc.*, **77**(396).

Burris, S. and Sankappanavar, H. P. (1981). *A Course in Universal Algebra*. Number 78 in Graduate Texts in Mathematics. Springer-Verlag, New York Heidelberg Berlin.

Chang, C. C. and Keisler, H. J. (1990). *Model Theory*, volume 73 of *Studies in logic and the foundations of mathematics*. North-Holland, Amsterdam - New York - Oxford - Tokyo, 3rd edition.

Groote, J. F. and Luttik, S. P. (1998). Undecidability and completeness results for process algebras with alternative quantification over data. Report SEN-R9806, CWI, The Netherlands. Available from `http://www.cwi.nl/~luttik/`; submitted for publication.

Groote, J. F. and Ponse, A. (1994a). Proof theory for μCRL: A language for processes with data. In D. J. Andrews, J. F. Groote, and C. A. Middelburg, editors, *Proceedings of the International Workshop on Semantics of Specification Languages*, Workshops in Computing, pages 232–251, Utrecht, The Netherlands. Springer-Verlag.

Groote, J. F. and Ponse, A. (1994b). The syntax and semantics of μCRL. In A. Ponse, C. Verhoef, and S. F. M. van Vlijmen, editors, *Algebra of Communicating Processes*, Workshops in Computing, pages 26–62, Utrecht, The Netherlands. Springer-Verlag.

Henkin, L., Monk, J. D., and Tarski, A. (1971). *Cylindric Algebras – Part I*, volume 64 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Company.

Henkin, L., Monk, J. D., and Tarski, A. (1985). *Cylindric Algebras – Part II*, volume 115 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Company.

Hennessy, M. (1991). A proof system for communicating processes with value-passing. *Formal Aspects of Computing*, **3**, 346–366.

McKenzie, R. N., McNulty, G. F., and Taylor, W. F. (1987). *Algebras, Lattices, Varieties — Volume I*. Wadsworth & Brooks/Cole, Monterey, California.

Meinke, K. and Tucker, J. V. (1992). Universal algebra. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Background: Mathematical Structures*, volume 1 of *Handbook of Logic in Computer Science*, pages 189–411. Oxford Science Publications.

Milner, R. (1983). Calculi for synchrony and asynchrony. *Theoret. Comput. Sci.*, **28**(3), 267–310.

Milner, R., Parrow, J., and Walker, D. (1992). A calculus of mobile processes, i. *Inform. and Comput.*, **100**, 1–40.

Monk, D. (1965). Substitutionless predicate logic with identity. *Archiv für Mathematische Logik und Grundlagenforschung*, **7**, 102–121.

Parrow, J. and Victor, B. (1998). The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS'98*, pages 176–185. IEEE Computer Society Press.

Shoenfield, J. R. (1967). *Mathematical Logic*. Addison-Wesley Publishing Company.

Tarski, A. (1965). A simplified formalization of predicate logic with identity. *Archiv für Mathematische Logik und Grundlagenforschung*, **7**(3–4), 61–79.