ISSN 1361 - 6161

THE UNIVERSITY of MANCHESTER

Computer Science
University of Manchester

Notes on P-Algebra (1): Process Structures

Kohei Honda

Department of Computer Science

University of Manchester

Technical Report Series

UMCS-95-12-2

Notes on P-Algebra (1): Process Structures*

Kohei Honda

Department of Computer Science
University of Manchester
Oxford Road, Manchester, UK.
khonda@cs.man.ac.uk

December 17, 1995

---

*Refereed by: Cliff Jones

Abstract

We develop an elementary theory of *process structure*, a mathematical object which is intended to capture a basic aspect of concurrent processes in a simple way. It has two kinds of presentations: one underlies many extant process theories and another has a more geometric flavour. We prove these are essentially equivalent. The theory forms a basis of P-algebra, a general semantic structure for concurrent processes.

1 Introduction

The aim of the present notes is to develop an elementary theory of *process structure*, a mathematical object which is intended to capture a basic aspect of concurrent processes in a simple way. It has two kinds of presentations: one underlies many extant process theories and another has a more geometric flavour. We prove there are essentially equivalent. The theory forms a basis of P-algebra, a general semantic structure for concurrent processes, to be developed in the sequels to the present notes.

While we leave discussions on the overall motivations underlying the study of process structures and P-algebra to elsewhere, let us comment briefly on what *this* exposition achieves. Processes in process algebras usually arise as (congruence classes of) terms in some term algebra, and, there, the idea of names, either for communication ports or for actions, plays an important role. So here are two entities, *processes* and *names*. We first form a class of simple algebras based on only these two elements, and dig out some structures hidden under their simple outlook. The observations now lead to another presentation of the same notion, simpler and more intuitive, which somehow dispenses with names. A theory of nameless processes is then developed, which can do the same thing as the original one, though with a quite different outlook. We finally prove the equivalence of the two theories in terms of both individual objects and their ambient categorical universes.

The study in the present notes can be regarded as part of our trial to delineate the notion of processes as a pure mathematical object, and to develop a general theory around it. The subsequent expositions will hopefully clarify what can be done in this vein. Discussions on other related theories including Abramsky's *Interaction Categories* [1], Berry and Boudol's *Chemical Abstract Machine* [2], Girard's *Linear Logic* [5], Meseguer's *Concurrent Term Rewriting Systems* [10], and Milner's *Action Structures* [14] will also be given in the future expositions, both in terms of their general orientation and, if possible, in terms of their concrete mathematical relationship to the present work. The aim of the the overall framework lies in finding a clean mathematical abstraction for structural and dynamic aspects of computation. The relationship to the works by Girard starting from his Linear Logic [5] in this context will also be discussed in future occasions. In terms of general orientation, it should be recorded that Cliff Jones strongly suggested to the author that names could be abstracted away from concurrent combinators [7, 8]. While the major part of the present theory, together with Yoshida's construction in [16], had already been developed at the time, his comment surely accelerated its formulation.

In the rest of the paper, Chapter 2 studies "rooted" process structures, i.e. the presentation which use names. Chapter 3 studies the non-rooted, or proper, process structures, and shows that they are equivalent to the first presentation.

One idea from group theory is extensively used in the notes, which we list in the following for reference.

1.1 Definition. Let G be a group with the identity e and X be a set. Then a *group action of* G *on* X is a function $* : G \times X \to X$ such that:

(i) $*(e, x) = x$ for any $x \in X$.

(ii) $*(g_1 g_2, x) = *(g_1, *(g_2, x))$ for any $g_1, g_2 \in G$ and for any $x \in X$.

We often write gx instead of $*(g, x)$, and say X is a G*-set* under the group action.

Said differently, a group action is an algebra over X with operations from the elements of a group G such that $ex = x$ and $(g_1 g_2)x = g_1(g_2 x)$ for any $g_1, g_2 \in G$ and $x \in X$. Yet another way of saying the same thing is that a group action is a homomorphism from a group G to the group of permutations over X.

Other related notions will be introduced as we go on. The author thanks Takayasu Ito, Cliff Jones, and Aki Yonezawa for their suggestions and encouragements, Nobuko Yoshida for discussions and nice pictures. I especially thank Cliff Jones for weekly discussions on this and related subjects which have always been enlightening to me. This work is supported by EPSRC fellowships.

Notes. The main result of the present paper first appeared in: Notes on P-Algebra (1): Process Strcuture, *Proc.TPPP'94*, LNCS 907, Spinger-Verlag, 1995. Some proofs are refined and typos are corrected in this new version.

2 Rooted Process Structure

Rooted process structures are a class of algebras which we can more or less directly extract from many extant process algebras, acting as a bridge from the essentially syntactic notion of processes to the one developed in Chapter 3. After reviewing basic algebraic elements, the idea of symmetry of processes is studied, which anticipates the construction in Chapter 3.

2.1 Definition. (rooted process structures) Given a countable set $\mathcal{N}$ of names, ranged over by $a, b, c, \ldots$, an $\mathcal{N}$*-rooted process structure* or just a *rooted process structure* is given by a set $\mathcal{P}$ of rooted processes, ranged over by $P, Q, R, \ldots$, together with the following data:

(i) An operation $[\sigma] : \mathcal{P} \to \mathcal{P}$, written postfix, for each bijection σ over $\mathcal{N}$. $[\sigma]$ is called *name permutation by* σ.

(ii) A function $\mathcal{FN} : \mathcal{P} \to 2^{\mathcal{N}}$, assuming $\mathcal{FN}(P)$ is always finite. $\mathcal{FN}(P)$ is called *free names* of P.

(iii) Equations:

1. $P[\sigma][\sigma'] = P[\sigma' \circ \sigma]$ (here $\circ$ is a functional composition).
2. $(a \in \mathcal{FN}(P) \;\; \Rightarrow \;\; \sigma(a) = a) \;\; \Rightarrow \;\; P[\sigma] = P$.
3. $\mathcal{FN}(P[\sigma]) = \sigma(\mathcal{FN}(P))$.

We usually fix some $\mathcal{N}$, and let $A, A', \ldots$ or $A_{\mathcal{P}}, A'_{\mathcal{P}'}, \ldots$ range over rooted process structures.

In other words, a rooted process structure $A_{\mathcal{P}}$ is a group action of the group of permutations over $\mathcal{N}$ on $\mathcal{P}$ (cf. 2.5 (i)), equipped with an extra function $\mathcal{FN}(\cdot)$, relating the permutations and $\mathcal{P}$. Note, as an algebra, $\mathcal{N}$ is *not* a constituent of a rooted process structure, but rather its prerequisite, since the signature depends on $\mathcal{N}$.

Some remarks on Definition 2.1:

2.2 Discussions.

(1) A rooted process P denotes an entity having several interface points with the outside, which are connected to (or *rooted to*) names, given by $\mathcal{FN}(P)$. Such a structure is often implicit in process algebras, in some ways or others, and notions like dynamics of interaction and process composition are formulated on its basis. We will see that this structure is worth studying as a separate mathematical object.

(2) Explicit introduction of *bijective renaming* as operation is essential. It acts on processes and relate them by name correspondence. The idea should be strictly differentiated from non-injective renaming (such as usual substitution), whose introduction necessitates another level of mathematical abstraction, essentially P-algebra itself.

(3) The free name function makes the range of names definite which a process is rooted to and which, therefore, bijective renaming has real effect on. This has a far reaching consequence in representation of rooted processes as will be studied in this section. We note that other formulations of the idea are possible, cf. 2.5 (iii).

2.3 Examples.

(i) Terms of CCS [11], possibly modulo strong bisimilarity, give such a structure, forgetting various constructors for process composition and augmented with $[\sigma]$ (defined syntactically) and $\mathcal{FN}(P)$ (using the static/dynamic sorts). It would be better to assume that $\bar{a}.P$ denotes a constructor which is distinct from $a.P$ but using the same name a (instead of having a single prefix for a name and its coname), thus e.g. $a.0$ and $\bar{a}.0$ have the same set of free names, $\{a\}$. Similar comments apply to terms of CSP [?].

(ii) Terms of π-calculus [15] or ν-calculus [6] modulo α-convertibility, or better, modulo structural equality [2, 13], augmented with $[\sigma]$ and $\mathcal{FN}(P)$. In fact a prototypical example of a rooted process structure is the combinatory representation of these calculi studied in [7, 8].

(iii) Another example is the set of terms in the pure λ-calculus modulo α-convertibility, where we understand the variables occurring free in a term as its free names.

We now present some immediate consequences of Definition 2.1.

2.4 Notations. By $S_{\mathcal{N}}$ we mean the group of permutations over $\mathcal{N}$, where the group operation is given by permutation multiplication, written $\sigma \circ \sigma'$ etc. We write $id_{\mathcal{N}}$ for the identity of $S_{\mathcal{N}}$, and σ^{-1} for the inverse of σ.

2.5 Proposition.

(i) *A rooted process structure $A_{\mathcal{P}}$ forms a group action of $S_{\mathcal{N}}$ on $\mathcal{P}$.*

(ii) *With $A \subset \mathcal{N}$, let $\sigma \upharpoonright A$ denote a function restricted to the domain A. Then for any P, σ_1 and σ_2, we have:*

$$\sigma_1 \upharpoonright A = \sigma_2 \upharpoonright A \ \wedge\ \mathcal{FN}(P) \subset A \quad \Rightarrow \quad P[\sigma_1] = P[\sigma_2]\ .$$

(iii) *Call $A \subset \mathcal{N}$ a* base *of P if, for any σ_1 and σ_2,*

$$\sigma_1 \upharpoonright A = \sigma_2 \upharpoonright A \quad \Rightarrow \quad P[\sigma_1] = P[\sigma_2]\ .$$

Then $\bigcap\{A | A \textit{ is a base of } P\} = \mathcal{FN}(P)$.

PROOF: For (i), we should show two conditions in Definition 1.1 are satisfied. But the condition (ii) of 1.1 is just Definition 2.1.(iii)-1, while the condition (i), i.e. $P[id_{\mathcal{N}}] = P$, follows from Definition 2.1.(iii)-2. For (ii), note if $a \in A$ then $\sigma_2^{-1} \circ \sigma_1\ (a) = a$, hence $P[\sigma_1][\sigma_2^{-1}] = P$ by Definition 2.1 (iii)-2, which is equivalent to $P[\sigma_1] = P[\sigma_2]$, as required. But this says $\mathcal{FN}(P)$ itself is a base of P, and if $A \subsetneq \mathcal{FN}(P)$ cannot be the base, so that (iii) holds. ■

Note (iii) suggests an alternative way of formulating the idea of free names (we further require that an arbitrary intersection of bases is again a base).

As any study of algebraic structure, we start with relations and maps over rooted processes.

2.6 Definition. (compatible relations and homomorphisms) A *compatible relation* from $A_{\mathcal{P}}$ to $A'_{\mathcal{P}'}$ is the one so with respect to operations $\{[\sigma]\}$, i.e. a relation $\mathcal{R}$ from $\mathcal{P}$ to $\mathcal{P}'$ such that $P\mathcal{R}Q \ \Rightarrow\ P[\sigma]\mathcal{R}Q[\sigma]$ for any $[\sigma]$. A compatible relation which is also an equivalence relation is a *congruence*. A compatible relation which is also a function is a *homomorphism*. *Isomorphisms* are bijective homomorphisms. We write $A \simeq A'$ if there is an isomorphism between A and A'.

2.7 Discussions. (compatible relation) Relations closed under name permutation are omnipresent in process algebra, e.g. bisimilarities and other standard equivalences and preorders, reduction relation over process terms, and, with a slight extension, labeled transition relation, etc. More deeply, this is because the essential use of names in relating processes lies in *specification of structural correspondence on interface points between processes*. Take an example of CCS, $a.b.0 + b.a.0 \sim a.0|b.0$, with $\sim$ being a strong bisimilarity. Here a and b are significant precisely because they specify how these two processes are structurally related. So *as far as the same correspondence is maintained*, we can permute names freely, e.g. $l.m.0 + m.l.0 \sim l.0|m.0$ holds again.

As in usual algebra (cf. [3]), compatible relations and quotient sets can be considered as assuming the same structure as the underlying object.

2.8 Proposition.

(i) *Given a compatible relation* $\mathcal{R} \stackrel{\text{def}}{=} \{\langle P,\ Q\rangle\}$*, take* $\mathcal{R}$ *itself as the set of processes and define:* (a) $\langle P,\ Q\rangle[\sigma] \stackrel{\text{def}}{=} \langle P[\sigma],\ Q[\sigma]\rangle$ *and* (b) $\mathcal{FN}(\langle P,\ Q\rangle) = \mathcal{FN}(P) \cup \mathcal{FN}(Q)$. *Then* $\mathcal{R}$ *forms a rooted process structure.*

(ii) *Given a congruence* $\sim$ *on* A*, write* $[P]_\sim$ *for an equivalence class by* $\sim$*. Define:* (a) $[P]_\sim[\sigma] \stackrel{\text{def}}{=} [P\sigma]_\sim$ *and* (b) $\mathcal{FN}([P]_\sim) \stackrel{\text{def}}{=} \bigcap_{P' \sim P} \mathcal{FN}(P')$. *Then* $\mathcal{P}/\sim$ *forms a rooted process structure. We call this structure the* quotient *of* A *by* $\sim$ *and write it* $A/\sim$.

PROOF: In both (i) and (ii), the property as algebra is standard, so we check consistency of the definition of free names with equations 2 and 3 of Definition 2.1.(iii). For tuples, suppose, for σ, $a \in \mathcal{FN}(\langle P,\ Q\rangle)$ implies $\sigma(a) = a$. Then it is identity on both $\mathcal{FN}(P)$ and $\mathcal{FN}(Q)$, hence $\langle P,\ Q\rangle[\sigma] = \langle P,\ Q\rangle$, therefore the second equation is satisfied. As to the third equation, $\mathcal{FN}(\langle P,\ Q\rangle[\sigma]) = \mathcal{FN}(P[\sigma]) \cup \mathcal{FN}(Q[\sigma]) = \sigma(\mathcal{FN}(P)) \cup \sigma(\mathcal{FN}(Q)) = \sigma(\mathcal{FN}(P) \cup \mathcal{FN}(Q))$, hence done.

For equivalence classes, let $A = \bigcap\{\mathcal{FN}(P') | P' \in [P]_\mathcal{R}\}$. Suppose $\sigma_1 \restriction A = \sigma_2 \restriction A$. Then we get: $P[\sigma_1] = P \prod_{1 \le i \le n}[\binom{\sigma_2(x_i)\sigma_1(b_i)}{\sigma_1(b_i)\sigma_2(x_i)}] \circ [\sigma_2] \circ \prod_{1 \le i \le n}[\binom{b_i x_i}{x_i b_i}]$ where $\{b_i\} = \mathcal{FN}(P) \backslash A$ and $\{x_i\}$ are taken fresh such that $\sigma_2(x_i) \notin \sigma_1(\mathcal{FN}(P))$. If, now, $\{b_i\}$ is *not* empty, there is a name b_1 which is *not* in $\mathcal{FN}(P')$ for some $P' \in [P]_\mathcal{R}$. By compatibility and because we have $P'[\binom{b_i x_i}{x_i b_i}] = P'$ for such P', we have:

$$\begin{array}{rcl} P[\rho_1] & \mathcal{R} & P' \prod_{1 \le i \le n}[\binom{\sigma_2(x_i)\sigma_1(b_i)}{\sigma_1(b_i)\sigma_2(x_i)}] \circ [\sigma_2] \circ \prod_{1 \le i \le n}[\binom{b_i x_i}{x_i b_i}] \\ & = & P' \prod_{2 \le i \le n}[\binom{\sigma_2(x_i)\sigma_1(b_i)}{\sigma_1(b_i)\sigma_2(x_i)}] \circ [\sigma_2] \circ \prod_{2 \le i \le n}[\binom{b_i x_i}{x_i b_i}]. \end{array}$$

Repeating the procedure for each b_i and using transitivity of $\mathcal{R}$, we finally get $P[\sigma_1]\mathcal{R}P[\sigma_2]$, i.e. $[P]_\mathcal{R}[\sigma_1] = [P]_\mathcal{R}[\sigma_2]$, as a consequence of $\sigma_1 \restriction A = \sigma_2 \restriction A$. Therefore, if for any $a \in \bigcap\{\mathcal{FN}(P') | P' \in [P]_\mathcal{R}\}$, $\sigma(a) = a$ holds, then the above discussions show $[P]_\mathcal{R}[\sigma] = [P]_\mathcal{R}[id_\mathcal{N}] = [P]_\mathcal{R}$. Regarding the third equation, it holds since $\mathcal{FN}([P]_\mathcal{R}[\sigma]) =$

$\bigcap\{\sigma(\mathcal{FN}(P'))|P' \in [P]_{\mathcal{R}}\} = \sigma(\mathcal{FN}([P]_{\mathcal{R}}))$, hence done. ■

Next we turn to homomorphisms. In Definition 2.6, a homomorphism only considers operations $\{[\sigma]\}$, not the free name function (so it is just a group action morphism between $S_{\mathcal{N}}$-sets). The following shows that we can somehow recover the information a posteriori.

2.9 Proposition. *Given a homomorphism $\mathcal{F}$, suppose Q is in its image. Then we have: $\mathcal{FN}(Q) = \bigcap_{Q=\mathcal{F}(P')} \mathcal{FN}(P')$.*

PROOF: Suppose $Q = \mathcal{F}(P)$. If $a \in \mathcal{FN}(Q)\backslash\mathcal{FN}(P)$ then $\mathcal{FN}(Q[\binom{af}{fa}]) \neq \mathcal{FN}(Q)$ (f fresh) by Definition 2.1 (here $\binom{af}{fa}$ is understood as its *extension*, i.e. a permutation over $\mathcal{N}$ which includes $\binom{af}{fa}$ and is identity on $\mathcal{N}\backslash\{a, f\}$), but $Q[\binom{af}{fa}] = \mathcal{F}(P[\binom{af}{fa}]) = \mathcal{F}(P) = Q$, contradiction. But if $a \in \mathcal{FN}(P)\backslash\mathcal{FN}(Q)$ then $\mathcal{F}(P[\binom{af}{fa}]) = Q[\binom{af}{fa}] = Q$ (f fresh) with $a \notin P[\binom{af}{fa}]$ hence done. ■

Note how the above result is consistent with the definition of free names for the quotient construction in 2.8 (ii). This ensures that a natural homomorphism from A to $A/\sim$ to behave nicely with respect to free name function, and, conversely, any homomorphic image of A to be not only isomorphic to a certain quotient of A but have essentially the same free name function, making, as a whole, a coherent algebraic theory.

For isomorphisms, we can have a stronger statement.

2.10 Corollary. *If $\mathcal{F}$ is an isomorphism, we have $\mathcal{FN}(P) = \mathcal{FN}(\mathcal{F}(P))$.*

PROOF: Immediate from Proposition 2.9. ■

By the corollary, we know isomorphisms determine not only how group action operates on processes but also free names they own (later we show they further determine another essential internal structure of rooted processes, called symmetries). So in most imaginable purposes treating rooted process structures modulo isomorphisms is enough, and it is technically convenient that these properties are automatically ensured by establishing only group action isomorphisms.

We also mention one characterisation of a homomorphism, suggestive to its alternative presentation in Chapter 3. It roughly says that in which way you may look at the source process, a homomorphism will map it in a coherent way.

2.11 Proposition. *A compatible relation $\mathcal{R}$ is a homomorphism if and only if $P\mathcal{R}Q$ and $P[\sigma]\mathcal{R}Q'$ implies $Q' = Q[\sigma]$.*

PROOF: Take $\sigma = id_{\mathcal{N}}$ for "if" direction, while "only if" part comes from existence of inverse of σ. ■

We now move to the study of deeper mathematical structure of rooted processes. For the purpose, several group theoretic notions are essential, which we briefly summarise in the following.

2.12 Definition. Below let G be a group and X be a set, and consider a group action of G on X. We write gx for the action of g on x.

(i) (morphism) Given two G-sets X and Y, a *morphism* from X to Y is a function $F : X \to Y$ such that $F(gx) = g(Fx)$ for any $g \in G$ and $x \in X$. A morphism is an *isomorphism* if it is bijective.

(ii) (orbits) A maximal subset Y of X such that, for any $y_1, y_2 \in Y$, we have $y_2 = gy_1$ for some g, is called an *orbit* of Y. Orbits partition X and thus induce an equivalence relation.

(iii) (transitive G-set) A G-set X is *transitive* if for any $x_1, x_2 \in X$, we can write $x_2 = gx_1$. So, in (ii), each orbit is a transitive G-set, and X as a whole is a disjoint union of these transitive G-sets.

(iv) (conjugate) Given a subgroup G' of G, $G'' = gG'g^{-1}$ $(= \{gg'g^{-1} | g' \in G'\})$ is another subgroup of G. Then we say G' and G'' are *conjugate*. Note conjugate subgroups are isomorphic.

(v) (isotropy) Let G_x be $\{g \mid g \in G \text{ s.t. } gx = x\}$. Then G_x forms a subgroup for each $x \in X$, called *isotropy subgroup* of x. Note if $y = gx$ we have $G_y = gG_xg^{-1}$, i.e. G_x and G_{gx} are always conjugate.

For us the following result regarding isotropy is essential.

2.13 Proposition. *Two G-sets X and Y are isomorphic if and only if the following holds: there is a bijective correspondence between their orbits and, moreover, in each pair of orbits $\langle X_i, Y_j \rangle$ under this correspondence, we have $G_x = gG_yg^{-1}$ for some $x \in X_i$, $y \in Y_j$ and $g \in G$.*

PROOF: While this is an easy corollary of the well-known representation theorem of a G-set by left coset G-sets, let us nonetheless give a direct proof, which is short and straightforward. Suppose X' and Y' are transitive G-sets, and for some $x' \in X'$ and $y' \in Y'$, we have $G_x = g \cdot G_y \cdot g^{-1}$. It is standard that if $y'' = gy' \in Y'$ then $G_{y''} = g \cdot G_y \cdot g^{-1} = G_x$, so that, for any g, we have $G_{gx'} = G_{gy''}$ by Definition 2.12 (iii). Now construct a function F s.t. $F(gx') = g(Fx')$ and $F(x') = y''$ for any g. This is well-defined, because: if $g_1x' = g_2x'$ then $g_1x' = (g_2g_1^{-1}g_1)x'$ so that $g_2g_1^{-1} \in G_{g_1x'} = G_{g_1y''}$, therefore

$$F(g_1x') = g_1y'' = (g_2g_1^{-1}g_1)(y'') = g_2y''.$$

So F defines a map, and indeed a morphism. But if $F(g_1x') = F(g_2x')$ then $g_2g_1^{-1} \in G_{g_1y''} = G_{g_1x'}$ so that $g_1x' = (g_2g_1^{-1}g_1)(x') = g_2x'$, so that F is bijective hence an isomorphism. Returning to Proposition above, since each orbit is transitive, the bijective correspondence in the statement of the above proposition induces isomorphisms between respective orbits, which together form an isomorphism as a whole, hence done. ∎

2.14 Notations.

(i) In a rooted process structure, we write $P \overset{\mathrm{N}}{\sim} Q$ if P and Q are in the same orbit, i.e. if $P = Q[\sigma]$ for some $\sigma \in S_{\mathcal{N}}$. So the equivalence classes from $\overset{\mathrm{N}}{\sim}$ are orbits in $\mathcal{P}$ under $S_{\mathcal{N}}$.

(ii) In a rooted process structure, the isotropy group of P under $S_{\mathcal{N}}$ is written $S_{\mathcal{N}\langle P\rangle}$. Note, as noted above, $S_{\mathcal{N}\langle P[\rho]\rangle}$ is given by $\rho \cdot S_{\mathcal{N}\langle P\rangle} \cdot \rho^{-1}$, which is isomorphic to $S_{\mathcal{N}\langle P\rangle}$.

Now we introduce the key notion for our study of rooted process structures.

2.15 Definition. (symmetries) A *symmetry of* P is a permutation of $\mathcal{FN}(P)$ which, seen as an element of $S_{\mathcal{N}}$ by extension, is in $S_{\mathcal{N}\langle P\rangle}$. The set of symmetries of P form a group denoted by $sym(P)$.

Symmetry arises inevitably in processes:

2.16 Examples.

(1) In CCS, a permutation $\binom{ab}{ba}$ is a symmetry of a process "$a.0|b.0$," seen modulo strong bisimilarity.

(2) Similarly, "$a(x).(\bar{b}x|\bar{c}x)$" in π-calculus modulo structural equality has a symmetry $\binom{abc}{acb}$. As seen, symmetry in processes typically occurs when an agent contains a subterm of the form $P|(P\sigma)$ modulo commutative monoid law, cf. Berry and Boudol [2]. In contrast, λ-terms (viewed as processes, cf. Examples 2.3) only own trivial symmetry; the non-commutative nature of applicative structure is well known.

(3) Not being mentioned explicitly as such, the significance of symmetry in reasoning about processes *has* indeed been perceived in the literature. See e.g. Chapter 4 of [12] where it is used to reduce the size of a bisimulation.

One useful fact about isotropy and symmetry follows.

2.17 Proposition. *Given any isotropy $\rho \in S_{\mathcal{N}\langle P\rangle}$, its restriction (of the domain) to $\mathcal{FN}(P)$ gives a permutation of $\mathcal{FN}(P)$, which is indeed a symmetry of P.*

PROOF: But if ρ is not a bijection over $\mathcal{FN}(P)$, then $\rho(\mathcal{FN}(P)) \neq \mathcal{FN}(P)$, a contradiction. Then use Proposition 2.4 (ii). ∎

Symmetries give essential insights about our algebraic constructions we already presented.

2.18 Proposition.

(i) Let $\mathcal{R}$ be a compatible relation. By Proposition 2.8, $\mathcal{R}$ is a rooted process structure. Now suppose $\langle P,\ Q\rangle \in \mathcal{R}$. Then $sym(\langle P,\ Q\rangle) = \{\sigma \upharpoonright \mathcal{FN}(\langle P,\ Q\rangle \mid \sigma(P) = P \ \wedge\ \sigma(Q) = Q)\}$.

(ii) Let $\sim$ be a congruence and let $\mathcal{P}/\sim$ be the corresponding quotient, seen as rooted processes as in Proposition 2.8. Then the symmetries of processes are given by: $sym([P]_\sim) \stackrel{\text{def}}{=} \{\sigma \upharpoonright \mathcal{FN}([P]_\sim) \mid \sigma(P) \sim P\}$.

(iii) *Let* $\mathcal{F}$ *be a homomorphism. Then* $Q = \mathcal{F}(P)$ and $\rho \in sym(P)$ *implies* $\rho \upharpoonright \mathcal{FN}(Q) \in sym(Q)$. *If, moreover,* $\mathcal{F}$ *is an isomorphism, we have* $sym(P) = sym(Q)$.

PROOF: By Proposition 2.17, we have only to check that the specified permutations are isotropies. In (i), given such σ, we have $\langle P,\ Q\rangle[\sigma] = \langle P[\sigma],\ Q[\sigma]\rangle = \langle P,\ Q\rangle$. For (ii), if $P[\sigma] \sim P$, then $[P]_\sim$ and $[P]_\sim[\sigma] \stackrel{\text{def}}{=} [P[\sigma]]_\sim$ define the same set, hence done. For (iii), note we already know $\mathcal{FN}(Q) \subset \mathcal{FN}(P)$ by Proposition 2.9. But if $\rho \in sym(P)$ we have $Q[\rho] = \mathcal{F}(P[\rho]) = \mathcal{F}(P) = Q$ (here ρ is taken as its extension), therefore, by Proposition 2.17, it includes the symmetry of Q. Finally if $\mathcal{F}$ is an isomorphism, Proposition 2.10 tells us $\mathcal{FN}(P) = \mathcal{FN}(Q)$, hence $sym(P) \supset sym(Q)$. By taking $\mathcal{F}^{-1}$ we know $sym(P) \subset sym(Q)$. ■

Note:

(i) In (i), a symmetry of $\langle P,\ Q\rangle$ should come from one symmetry of P and another of Q whose values should coincide at names $\mathcal{FN}(P) \cap \mathcal{FN}(Q)$. In this case, therefore, symmetries decrease in general.

(ii) In contrast, in (ii) above, notice that we only require P and $P[\sigma]$ are equal up to the congruence. So even if no symmetry except identity exists originally, new symmetry would arise after quotienting. Thus *quotient construction may increase symmetries of processes.*[1]

(iii) In (iii), we know homomorphisms generally increase symmetries (in accordance with its close connection to quotient constructions). In addition, we now know that isomorphisms completely characterise symmetries in addition to group actions and free names, cf. 2.5.

Having seen how symmetries arise in rooted processes, we ask the following question: can symmetries completely replace isotropies, the power of the latter is already known by Proposition 2.13? The free name function works.

[1]Since the quotient may in general "cut off" unrelated names (typically, in process algebras, those which are semantically insignificant), the sheer number of symmetries may decrease; however the original symmetries are all inherited within the reduced set of names.

2.19 Lemma. (representation of isotropy) $S_{\mathcal{N}\langle P\rangle} = S_{\mathcal{N}\langle Q\rangle}$ *if and only if* $sym(P) = sym(Q)$.

PROOF: From Proposition 2.17, "if" direction is immediate. For "only if" direction, if we know that $S_{\mathcal{N}\langle P\rangle} = S_{\mathcal{N}\langle Q\rangle}$ implies $\mathcal{FN}(P) = \mathcal{FN}(Q)$, we get $sym(P) = sym(Q)$ (again) by Proposition 2.17. But suppose not, i.e. for some a we have e.g. $a \in \mathcal{FN}(P) \backslash \mathcal{FN}(Q)$, assuming $S_{\mathcal{N}\langle P\rangle} = S_{\mathcal{N}\langle Q\rangle}$. Take f fresh, then $\binom{af}{fa}$ is in $S_{\mathcal{N}\langle Q\rangle}$ but not in $S_{\mathcal{N}\langle P\rangle}$ by Definition 2.1 (iii)-3, which is a contradiction. ■

The main message of Lemma 2.19 is that, in rooted process structure, *symmetries characterise everything* (up to isomorphism). Thus, as the main result of the present section, we arrive at:

2.20 Proposition. (representation of a rooted process structure by symmetries) *Given* $A_{\mathcal{P}}$, *let* $I = \mathcal{P}/\overset{\mathrm{N}}{\sim}$, *the set of orbits in* $\mathcal{P}$. *Then a* symmetry presentation *of* $A_{\mathcal{P}}$, *denoted by* α, β, *etc. is an* I*-indexed family of symmetries, say* $\{sym(P_i)\}_{i\in I}$, *such that we have* $P_i \in i$ *for all* $i \in I$. *Then* α *and* β, *possibly from two different rooted process structures, are* isomorphic, *written* $\alpha \leftrightarrow \beta$, *when there is a bijection* F *between them such that:*

$$F(\alpha_i) = \sigma \cdot \alpha_i \cdot \sigma^{-1}$$

where α_i *denotes* i*'th member of* α. *Then* $\leftrightarrow$ *is an equivalence relation. Moreover:*

(i) *If* α *and* β *come from the same rooted process structure, then always* $\alpha \leftrightarrow \beta$.

(ii) *By* (i), *write* $A \leftrightarrow A'$ *when a pair of symmetry presentations of* A *and* A' *are isomorphic. Then we have* $A \leftrightarrow A'$ *if and only if* $A \simeq A'$.

PROOF: Since $P \overset{\mathrm{N}}{\sim} Q$ implies $S_{\mathcal{N}\langle P\rangle}$ and $S_{\mathcal{N}\langle Q\rangle}$ are conjugate, and, by Lemma 2.19, this implies the conjugacy of $sym(P)$ and $sym(Q)$ again, (i) follows. For (ii), "only if" direction follows from Proposition 2.13 and Lemma 2.19. For "if" direction, suppose $A_{\mathcal{P}} \simeq A'_{\mathcal{P}'}$, i.e. there is an isomorphism $\mathcal{F} : \mathcal{P} \to \mathcal{P}'$. Clearly each orbit in $\mathcal{P}$ is one-one mapped to an orbit in $\mathcal{P}'$, and, by Proposition 2.18 (iii), this map preserves symmetries, hence done. ■

For our later purpose, it is convenient to formulate the ambient universes of algebra of rooted processes as categorical structures. When dealing with relations, we also would like to preserve essential relational operations, thus mapping not only relations themselves but also relational reasoning to another domain.

2.21 Definition. A *complete boolean allegory* (cf. [4]) is a category in which, for each pair of objects A and B, we have:

(i) $hom(A, B)$ forms a complete boolean algebra, with operations $\bigcup, \bigcap, (\,\cdot\,)^{\mathcal{C}}$, the first two of which are compatible with categorical composition $\circ$, i.e. $(\bigcup_i R_i) \circ S = \bigcup_i (R_i \circ S)$ and $S \circ (\bigcup_i R_i) = \bigcup_i (S \circ R_i)$, similarly for $\bigcap$.

(ii) There is an operation $(\,\cdot\,)^{-1}$ which sends each element of $hom(A, B)$ to $hom(B, A)$ bijectively so that $R^{-1^{-1}} = R$, for which we have:

$$(R_1 \circ R_2)^{-1} = R_2^{-1} \circ R_1^{-1} \qquad (\bigcup_i R_i)^{-1} = \bigcup_i (R_i^{-1}) \qquad (R^{\mathcal{C}})^{-1} = (R^{-1})^{\mathcal{C}}$$

Functors between two complete boolean allegories are those between the underlying categories which also preserve these operations. Two complete boolean allegories are *equivalent as allegories* when they are categorically equivalent involving functors in the above sense.

Note, in (ii), $(\bigcap_i R_i)^{-1} = \bigcap_i (R_i^{-1})$ is derivable from the presented equations.

2.22 Proposition. (RPS_{rel} and RPS) *Fix some $\mathcal{N}$. Then the set of rooted process structures and name compatible relations between them form a complete boolean allegory. This allegory (as a category) contains the category of the same set and homomorphisms as its subcategory. We call these two structures,* RPS_{rel} *and* RPS*, respectively. Note isomorphisms coincide in* RPS_{rel} *and* RPS.

PROOF: For RPS_{rel}, we only take relational composition, inverse, and set union/intersection for the corresponding operations. ■

Some discussions on RPS and RPS_{rel}:

(1) Remember we take $\mathcal{N}$ as countable. Under this condition, usage of different sets of names only result in isomorphic allegories and categories. Thus it suffices to think of one pair of category and allegory.
(2) Clearly RPS is a full subcategory of all $S_{\mathcal{N}}$-sets and their morphisms, similarly for RPS_{rel}. This suggests a certain (but not much) generalisation of our results in these notes, though our present interest lies in these specific structures.
(3) In RPS, products are given by the set-theoretic products (seen as a rooted process structure as in Proposition 2.8), co-products by disjoint unions, the initial object by the empty set, and the final objects by structures with one element P such that $\mathcal{FN}(P) = \emptyset$. Note each object is given by a co-product of (a possibly infinite number of) structures each having only a single orbit.

2.23 Question. (leading to the next section) We already know that rooted processes have concise representation by a family of symmetries, cf. Proposition 2.20. These symmetry presentations are even equipped with an idea of isomorphism, but that is all, so we have not much other than the objects part of the corresponding categories above. What notions correspond to the *arrow parts* of RPS_{rel} and RPS? That is, how can we think of relations and maps between processes in the presentation by symmetries?

3 Process Structure and Equivalence Theorem

In the following we present a theory of processes which is essentially equivalent to what we have seen in the preceding Chapter but is based on more concise and direct constructs, exposing basic ideas of processes. In our study in the subsequent expositions, it will be shown how this presentation allows generalisation of various elements of process theories not possible or at least not amenable in the traditional formulations. Though having a close relationship to our development in Section 2 and remembering the question posed in the last subsection, we rather start with presenting the theory without referring to the preceding development, to make basic ideas come out as succinctly as possible. Later we will come back to the question on how two theories are related.

The following is the primary object of study in this Chapter.

3.1 Definition. (process structure) *A (non-rooted) process structure* is a triple $\langle \mathbf{P},\ \mathcal{H},\ \mathcal{S} \rangle$ where:

(i) $\mathbf{P}$ is the set of *(non-rooted, or pure) processes*, ranged over by $p, q, r, \ldots$.

(ii) $\mathcal{H}$ assigns, to each p, a set called *handles* of p. A handle is denoted by $h, h', \ldots$.

(iii) $\mathcal{S}$ assigns, to each p, a subgroup of $S_{\mathcal{H}(p)}$ (the group of permutations over $\mathcal{H}(p)$), called *symmetries on* p.

A *substructure* of a process structure is a subset of processes together with two assignments inherited from the original structure. The *arity* of a process, written $ar(p)$, is given by $|\mathcal{H}(p)|$, which we assume finite in the present paper. We sometimes write $p^{(n)}$ to denote the arity of p. Often the underlying set $\mathbf{P}$ denotes a process structure.

We can think of a process as a geometric object with multiple discrete points of connection and interaction:

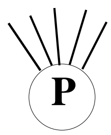

Figure 1: A Process

While such an object seems not to have been studied as such so far, some examples inducing similar structures can nevertheless be found.

3.2 Examples.

(1) Girard's Proof Nets [5], as well as Lafont's Interaction Nets [9], can be considered as forming process structures, regarding their "free edges" as handles.

(2) An important example is *process graphs* in [16], a graphical presentation of concurrent combinators [7, 8] (discussed in 2.3). In fact process graphs are the prototypical example of process structures, and form certain free objects in the universe of nameless processes. Symmetries come from those inherent in graphs.

In many senses, a process structure can be regarded as a generalisation of a set: instead of a collection of elements, we have a collection of processes, and the essential point is that we deal with a process considering its internal structures, i.e. its handles and symmetries. In essence, this comes from our foremost concern, i.e. to build a basic stratum of general process theory where we naturally have to consider dynamics and (in)equality among processes each of which has multiple interface points and interact with the outside through them. As a stratum, the present theory has no notion of composition of processes nor does it include direct formulation of such ideas as dynamics of process interaction and behavioural semantics. However it *does* provide the basic elements which would be essentially used for these purposes, such as theory of maps, relations, and quotients, each of which, in spite of quite different content, go surprisingly parallel to those counterparts in theory of sets and elements. We start with one general idea of how we can relate two processes. Soon we will be content with its less general variant, but if we remember only one way to relate two *elements* (in a set) is by forming a tuple, the following should be instructive.

3.3 Definition. (web) A *web from p_1 to p_2* is a triple $\langle p_1, \ \delta, \ p_2 \rangle$, where δ is a one-one partial map from a partition of $\mathcal{H}(p_1)$ to a partition of $\mathcal{H}(p_2)$.

A web relates two processes by mapping handles collection by collection. Note composable webs are composed to make another web and the inverse of a web is again such. Generality of webs in relating processes allows us to treat many interesting phenomena in concurrent processes. We however leave theory of webs to the subsequent exposition and concentrate on its special case, called *correspondences*, where in mapping a collection of handles to another collection, we restrict the collections to singletons, since this suffices for our present purpose.

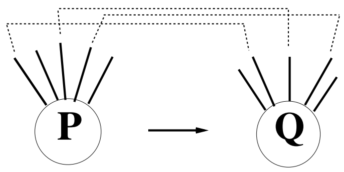

Figure 2: A Correspondence

3.4 Definition. (correspondence, cf. Figure 2)

(i) A *correspondence* is a triple $\langle p_1, \delta, p_2\rangle$ in which δ is injective but not necessarily total function from $\mathcal{H}(p_1)$ to $\mathcal{H}(p_2)$. By abuse of notation, we let $\delta, \delta, \ldots$ range over correspondences. $\mathtt{dom}(\delta)$ and $\mathtt{cod}(\delta)$ denote the source process and the target process, respectively. Then $\mathtt{pre}(\delta)$ is the handles for which δ is defined, while $\mathtt{im}(\delta)$ is the image of the component function in the handles of the target.

(ii) We say $\langle p, \delta, q\rangle$ and $\langle p, \delta', q\rangle$ are *essentially equivalent*, written $\delta \sim \delta'$, if, for some $\rho \in \mathcal{S}(p)$ and $\rho' \in \mathcal{S}(q)$, we have $\rho' \circ \delta \circ \rho = \delta'$. A correspondence is *surjective* (resp. *total*) if its component function is surjective (resp. total). $\mathbf{id}_p$ denotes an identity map on $\mathcal{H}(p)$.

(iii) Let $\langle p, \delta_1, q\rangle$ and $\langle q, \delta_2, r\rangle$ be correspondences. Then their *composition* is another correspondence $\langle p, \delta_2 \circ \delta_1, r\rangle$. The *inverse* of a correspondence $\langle p, \delta, q\rangle$ is $\langle q, \delta^{-1}, p\rangle$.

Correspondences are like *tuples*: then their collection forms what may be regarded as *relation* in the present setting.

3.5 Definition. (p-relation)

(i) A *p-relation* $\mathfrak{R}$ *from* $\mathbf{P}_1$ *to* $\mathbf{P}_2$ is a set of correspondences with domains in $\mathbf{P}_1$ and codomains in $\mathbf{P}_2$ such that if $\delta \in \mathfrak{R}$ and $\delta' \sim \delta$, then $\delta' \in \mathfrak{R}$ again. We write $p\mathfrak{R}_\delta q$ to mean p is related to q in $\mathfrak{R}$ by δ.

(ii) Given $\mathfrak{R}_1$ from $\mathbf{P}$ to $\mathbf{Q}$ and $\mathfrak{R}_2$ from $\mathbf{Q}$ to $\mathbf{R}$, their *composition*, written $\mathfrak{R}_2 \circ \mathfrak{R}_1$, is a p-relation from $\mathbf{P}$ to $\mathbf{R}$ given by: $\{\delta_2 \circ \delta_1 \mid \delta_1 \in \mathfrak{R}_1 \wedge \delta_2 \in \mathfrak{R}_2 \wedge \mathtt{dom}(\delta_2) = \mathtt{cod}(\delta_1)\}$. The *inverse* of $\mathfrak{R}$, written $\mathfrak{R}^{-1}$, is the set of inverse correspondences of the original one, again forming a p-relation. Given an arbitrary family $\{\mathfrak{R}_i\}_{i \in I}$ of p-relations all from $\mathbf{P}$ to $\mathbf{Q}$, their set-theoretic intersection and union are again p-relations, just written $\bigcap\{\mathfrak{R}_i\}_{i \in I}$ and $\bigcup\{\mathfrak{R}_i\}_{i \in I}$. Similarly for two p-relations sharing domain and codomain, their difference is again a p-relation, denoted by $\mathfrak{R} \backslash \mathfrak{R}'$.

(iii) The *product* of $\mathbf{P}_1$ and $\mathbf{P}_2$, written $\mathbf{P}_1 \times \mathbf{P}_2$, is the set of all correspondences between $\mathbf{P}_1$ and $\mathbf{P}_2$. The *identity p-relation* on $\mathbf{P}$ is the set of all symmetries (as correspondences) on processes of $\mathbf{P}$, written $\mathbf{ID_P}$. A p-relation $\mathfrak{R}$ on $\mathbf{P}$ (i.e. from $\mathbf{P}$ to $\mathbf{P}$) is *reflexive* if $\mathbf{ID_P} \subset \mathfrak{R}$, *transitive* if $\mathfrak{R} \circ \mathfrak{R} \subset \mathfrak{R}$, and *symmetric* if $\mathfrak{R}^{-1} = \mathfrak{R}$. If the three hold together, $\mathfrak{R}$ is an *equivalence p-relation.*

3.6 Discussions. (abstract correspondence) A basic building block of a p-relation is a correspondence modulo $\sim$ rather than a correspondence itself. Call such a correspondence an *abstract correspondence*: formally a p-relation is an abstract correspondence if it is non-empty and whenever δ and δ' are its elements we have $\delta \sim \delta'$. Note, from the viewpoint that correspondences relate two geometric objects, it is indeed a single abstract correspondence, not a correspondence, which represents a single relational situation between processes. In the light of the understanding, to formulate a p-relation as a collection of abstract correspondences is a natural idea, which gives an essential equivalent theory and moreover renders some constructions (like 3.10) more natural. However there is a subtle point in treating abstract correspondences, which would make the present formulation easier to grasp: composition of two abstract correspondences may *not* form another (single) abstract correspondence. To see this, given $p[h]$, $q[i_1, i_2]$, and $r[j]$ (here $[h]$ etc. shows handles a process owns) with $\mathcal{S}(q) = \{\binom{i_1 i_2}{i_2 i_1}, \binom{i_1 i_2}{i_1 i_2}\}$. Now if we compose $\{\langle p,\ \{h \mapsto i_1\},\ q\rangle, \langle p,\ \{h \mapsto i_2\},\ q\rangle\}$ and $\{\langle q,\ \{i_1 \mapsto j\},\ r\rangle, \langle q,\ \{i_2 \mapsto j\},\ r\rangle\}$, both being abstract correspondences, then we get: $\{\langle p,\ \{h \mapsto j\},\ q\rangle,\ \langle p,\ \emptyset,\ q\rangle\}$, which is the union of *two* abstract correspondences. This shows that the composition of p-relations needs to be understood with care.

The point we just noted is especially pertinent when we formulate function-like p-relations, as follows.

3.7 Definition. (p-map) A correspondence $\langle p,\ \delta,\ q\rangle$ is *symmetry preserving* when, if $\rho \in \mathcal{S}(p)$, for some $\rho' \in \mathcal{S}(q)$, we have $\delta \circ \rho = \rho' \circ \delta$. Then a *p-map* is a p-relation $\mathfrak{R}$ such that: (1) $\mathfrak{R}$ contains, for each process in the source, a set of correspondences any two of which are equated by $\sim$, and (2) $\delta \in \mathfrak{R}$ implies δ is surjective and symmetry preserving. A p-map whose inverse is also a p-map is called an *isomorphism*. By the *image* of a p-map $\mathfrak{R}$, we mean a sub process structure of $\mathtt{cod}(\mathfrak{R})$ given by $\{q \mid \exists p.\ p\mathfrak{R}_\delta q\}$.

Note a p-map consists of a collection of abstract correspondences one for each process in the domain, in the sense of 3.6. Note also we define isomorphisms to be stronger than bijective p-maps. By imposing the conditions as given above, p-maps indeed behave like functions between sets. For example, symmetry preservation has the following important consequence:

3.8 Proposition. *Given two p-maps composable as p-relations, their composition is again a p-map.*

PROOF: Let $\mathfrak{R}_1$ and $\mathfrak{R}_2$ be p-maps composable as p-relations. Then immediately $\mathfrak{R}_2 \circ \mathfrak{R}_1$ is a p-relation, its correspondences are all surjective, and it is everywhere defined in its source. Moreover if $\langle p,\ \delta_2 \circ \delta_1,\ q\rangle$ is a composed correspondence and if $\rho_1 \in \mathcal{S}(p_1)$, then $\delta_2 \circ \delta_1 \circ \rho_1 = \delta_2 \circ \rho_2 \circ \delta_1 = \rho_3 \circ \delta_2 \circ \delta_1$ where $\rho_2 \in \mathcal{S}(\mathtt{cod}(\delta_1)) = \mathcal{S}(\mathtt{dom}(\delta_2))$ and $\rho_3 \in \mathcal{S}(\mathtt{cod}(\delta_2))$, hence it is symmetry preserving. The rest is the uniqueness of the value modulo $\sim$. Suppose $\delta_2' \circ \delta_1'$ as well as $\delta_2 \circ \delta_1$ is in $\mathfrak{R}_2 \circ \mathfrak{R}_1$, whose domain coincides. But this means $\delta_1 \sim \delta_1'$ hence $\delta_2 \sim \delta_2'$, so we can write $\delta_1' = \rho_2 \circ \delta_1 \circ \rho_1$ and $\delta_2' = \pi_3 \circ \delta_1 \circ \pi_2$ where $\rho_{1,2}$ and $\pi_{1,2}$ are symmetries of respective processes. But using symmetry preservation, for an appropriate symmetry π_3':

$$\begin{array}{rcl} \delta_2' \circ \delta_1' & \stackrel{\text{def}}{=} & \pi_3 \circ \delta_2 \circ \pi_2 \circ \rho_2 \circ \delta_1 \circ \rho_1 \\ & = & \pi_3 \circ \pi_3' \circ \delta_2 \circ \delta_1 \circ \rho_1 \quad \sim \ \delta_2 \circ \delta_1 \end{array}$$

hence done. ■

Next we show how isomorphisms deserve their name, a consequence of surjectivity of correspondences in p-maps.

3.9 Proposition. (isomorphisms)

(i) *Let $\mathfrak{R}$ be an isomorphism. Then $\mathfrak{R}$ is bijective on processes. If $\delta \in \mathfrak{R}$, then δ is also bijective.*

(ii) *Let $\mathfrak{R}$ be an isomorphism and $\langle p,\ \delta,\ q\rangle \in \mathfrak{R}$. Then $\mathcal{S}(p)$ and $\mathcal{S}(q)$ are isomorphic in the way:*

(1) $$\mathcal{S}(q) = \theta \mathcal{S}(p) \theta^{-1}$$

where θ is some bijection from $\mathcal{H}(p)$ to $\mathcal{H}(q)$.

(iii) **P** *and* **Q** *are isomorphic if and only if there is a bijective correspondence between processes which induce, for each pair $\langle p,\ q\rangle$, a bijection on handles δ such that the equation* (1) *holds.*

Proof: (i) is immediate from the function-like characters of p-maps and surjectivity of correspondences in isomorphisms. (ii) follows from that, by (i) above, δ is a bijection and that symmetry preservation should be both-way in isomorphisms. Finally "only if" of (iii) is immediate, while, if there is a bijection δ between $\mathcal{H}(p)$ and $\mathcal{H}(q)$ as specified, then we can write, for any $\rho \in \mathcal{S}(p)$, $\rho = \delta^{-1} \circ \rho' \circ \delta$, that is, $\delta \circ \rho = \rho' \circ \delta$ for some $\rho' \in \mathcal{S}(q)$, that is, δ is symmetric preserving. Similarly any bijection of the form $\rho_2 \circ \delta \circ \rho_1$ where $\rho_1 \in \mathcal{S}(p)$ and $\rho_2 \in \mathcal{S}(q)$ is symmetry preserving. Collect all these bijections for each process, and the result is the required isomorphism. ■

By (i) and (ii), isomorphisms preserve all essential information in process structures. Note, without the condition on surjectivity, (i) (hence (ii)) does not hold. At the same time, one may notice there are some interests in studying non-surjective p-maps since, e.g. Proposition 3.8 does not depend on surjectivity. At any rate, surjectivity does result in a simpler theory, see how its combination with symmetry preservation results in a coherent notion of natural p-map later.

Relations can be treated as sets, and thus relational reasoning includes set operations, which is useful. Just like that, a p-relation is another process structure.

3.10 Proposition. (p-relation as process structure) *Given a p-relation $\mathfrak{R}$, define:*

(i) Processes: $\mathfrak{R}/\sim$. We take one correspondence from each equivalence class and use it for the equivalence class.

(ii) Handles: $\mathcal{H}(\langle p,\ \delta,\ q\rangle) \stackrel{\text{def}}{=} (\mathcal{H}(p)\backslash\mathtt{im}(\delta^{-1})) \uplus \delta \uplus (\mathcal{H}(q)\backslash\mathtt{im}(\delta))$.

(iii) $\mathcal{S}(\langle p,\ \delta,\ q\rangle) \stackrel{\text{def}}{=} \{\langle \rho_1,\ \rho_2\rangle \mid \rho_2^{-1}\circ\delta\circ\rho_1 = \delta,\ \rho_1 \in \mathcal{S}(p_1) \wedge \rho_2 \in \mathcal{S}(p_2)\}$ where $\langle \rho_1,\ \rho_2\rangle$ acts on $\mathcal{H}(\langle p,\ \delta,\ q\rangle)$ as:

$$\begin{array}{rcl}\langle \rho_1,\ \rho_2\rangle(h_1) & = & \rho_1(h_1) \quad (h_1 \in \mathcal{H}(p)\backslash\mathtt{pre}(\delta))\\ \langle \rho_1,\ \rho_2\rangle(h_1 \mapsto h_2) & = & \rho_1^{-1}(h_1) \mapsto \rho_2(h_2).\\ \langle \rho_1,\ \rho_2\rangle(h_2) & = & \rho_2(h_2) \quad (h_2 \in \mathcal{H}(q)\backslash\mathtt{im}(\delta).\end{array}$$

then the above data define a process structure. Moreover, different choice from an equivalence class only results in isomorphic process structures.

PROOF: As to the first statement, the only point to check is whether $\mathcal{S}(\langle p,\ \delta,\ q\rangle)$ really defines a symmetry, i.e. whether it forms a subgroup of $S_{\mathcal{H}(\langle p,\ \delta,\ q\rangle)}$. Now how does a tuple $\langle \rho_1,\ \rho_2\rangle$ operate on $\mathcal{H}(\langle p,\ \delta,\ q\rangle)$? At $\langle h_1,\ h_2\rangle \in \delta$, we get $\langle \rho_1(h_1),\ \rho_2(h_2)\rangle$. By (ii) we have $\delta(\rho_1(h_1)) = \rho_2(\delta(h_1))$ so $\langle \rho_1(h_1),\ \rho_2(h_2)\rangle \in \delta$, and the map is obviously injective, so it gives a permutation over δ. This also tells us that an orbit of ρ_1 containing $h \in \mathcal{H}(p)\backslash\mathtt{im}(\delta^{-1})$ can never intersect with $\mathtt{im}(\delta^{-1})$, similarly for ρ_2. So the operation as a whole becomes a permutation over $\mathcal{H}(\langle p,\ \delta,\ q\rangle)$. Since the identity and the inverse can easily be defined, we conclude that the symmetry is well-defined. As to the second statement, take $\langle p_1,\ \delta,\ p_2\rangle$ and $\langle p_1,\ \delta',\ p_2\rangle$, with $\delta' \stackrel{\text{def}}{=} \rho_2\circ\delta\circ\rho_1$ where $\rho_1 \in \mathcal{S}(p_1)$ and $\rho_2 \in \mathcal{S}(p_2)$. Then it is easy to show $\mathcal{S}(\langle p_1,\ \delta',\ p_2\rangle) = \langle \rho_1^{-1},\ \rho_2\rangle \mathcal{S}(\langle p_1,\ \delta,\ p_2\rangle)\langle \rho_1^{-1},\ \rho_2\rangle^{-1}$ where the composition is done componentwise. Note, moreover, that $\langle \rho_1^{-1},\ \rho_2\rangle$ gives a bijection from $\mathcal{H}(\langle p_1,\ \delta,\ p_2\rangle)$ to $\mathcal{H}(\langle p_1,\ \delta',\ p_2\rangle)$ in the way:

$$\begin{array}{rcl}\langle \rho_1^{-1},\ \rho_2\rangle(h_1) & = & \rho_1^{-1}(h_1) \quad (h_1 \in \mathcal{H}(p_1)\backslash\mathtt{pre}(\delta))\\ \langle \rho_1^{-1},\ \rho_2\rangle(h_1 \mapsto h_2) & = & \rho_1^{-1}(h_1) \mapsto \rho_2(h_2).\\ \langle \rho_1^{-1},\ \rho_2\rangle(h_2) & = & \rho_2(h_2) \quad (h_2 \in \mathcal{H}(p_2)\backslash\mathtt{im}(\delta)).\end{array}$$

Then we use Proposition 3.9 (iii). ■

How can we know that this definition, which is surely consistent, really "works"? The following shows one place where it does function well.

3.11 Proposition. (universality of product) *Given* $\mathbf{P}_1$ *and* $\mathbf{P}_2$, *define two p-maps* $\Pi_1 : \mathbf{P}_1 \times \mathbf{P}_2 \to \mathbf{P}_1$ *and* $\Pi_2 : \mathbf{P}_1 \times \mathbf{P}_2 \to \mathbf{P}_2$ *such that:*

(i) if $\langle\langle p_1,\ \delta,\ p_2\rangle,\ \pi,\ p_1\rangle \in \Pi_1$, we have $\pi : h \mapsto h$ for $h \in \mathcal{H}(p_1)\backslash\mathtt{im}(\delta^{-1})$ and $\pi : (h_1 \mapsto h_2) \mapsto h_1$ for $(h_1 \mapsto h_2) \in \delta$, else undefined.

(ii) Similarly for Π_2.

Then whenever we have two p-maps $\mathfrak{R}_1 : \mathbf{P}' \to \mathbf{P}_1$ *and* $\mathfrak{R}_2 : \mathbf{P}' \to \mathbf{P}_2$ *for some* $\mathbf{P}'$*, then there is a unique p-map* Ψ *such that* $\mathfrak{R}_1 = \Pi_1 \circ \Psi$ *and* $\mathfrak{R}_2 = \Pi_2 \circ \Psi$*.*

PROOF: That Π_1 and Π_2 are p-maps are easy (note by the definition of $\mathcal{S}(\langle p,\ \delta,\ q\rangle)$ in Proposition 3.10 both are obviously symmetry preserving). We leave the construction of Γ to the reader (hint: in the case $\langle p',\ \delta_1,\ p_1\rangle \in \mathfrak{R}_1$ and $\langle p',\ \delta_2,\ q_1\rangle \in \mathfrak{R}_2$, we say $\langle p_1,\ \delta,\ p_2\rangle$ is *consistent with* δ_1 *and* δ_2 when

$$\delta : h_1 \mapsto h_2 \quad \Leftrightarrow \quad \exists h' \in \mathcal{H}(p').\ (\delta_1 : h' \mapsto h \wedge \delta_2 : h' \mapsto h_2).$$

Construct a correspondence from p' to the above correspondence (qua process) as a component of Γ. Show that if another map is composed with Π_1 and Π_2, the required property cannot hold). ■

There are other coherence conditions which the construction of Proposition 3.10 satisfies, which we do not discuss. Now we move to the quotient construction. We need the following notions: Given an equivalence $\mathfrak{R}$, the set of $\mathfrak{R}$-*effective handles*, written $\mathcal{H}(p)|\mathfrak{R}$, is a set of elements of $\mathcal{H}(p)$ which are in $\mathtt{im}(\delta)$ for any $\delta \in \mathfrak{R}$ with $\mathtt{cod}(\delta) = p$. Note:

3.12 Fact. *If* $\mathfrak{R}$ *is an equivalence, then* $p\mathfrak{R}_\delta q$ *implies* $\sharp(\mathcal{H}(p)|\mathfrak{R}) = \sharp(\mathcal{H}(q)|\mathfrak{R})$ *where* $\sharp(A)$ *denotes the cardinality of* A*.*

This is because, if not, any partial injection from a smaller one to a larger one cannot cover the latter. The quotient construction follows.

3.13 Definition. (quotient)

(i) Given an equivalence p-relation $\mathfrak{R}$ on $\mathbf{P}$, the *equivalence class* of a process p with respect to $\mathfrak{R}$ is defined as:

$$[p]_{\mathfrak{R}} \stackrel{\text{def}}{=} \{q \mid \exists\delta.\ \langle p,\ \delta,\ q\rangle \in \mathfrak{R}\}.$$

(ii) A *quotient* of $\mathbf{P}$ with respect to $\mathfrak{R}$ is given by the following data:

- The set $\{[p]_{\mathfrak{R}} \mid p \in \mathbf{P}\}$.
- The function $\mathcal{H}$ which assigns, to each $[p]_{\mathfrak{R}}$, $\mathcal{H}(p')|\mathfrak{R}$, choosing a specific $p' \in [p]_{\mathfrak{R}}$.
- The function $\mathcal{S}$ which assigns, to each $[p]_{\mathfrak{R}}$, a set of permutations $\mathcal{S}([p]_{\mathfrak{R}})$ over $\mathcal{H}([p]_{\mathfrak{R}})$, such that, with p' being a process chosen from $[p]_{\mathfrak{R}}$:

$$s \in \mathcal{S}([p]_{\mathfrak{R}}) \quad \Leftrightarrow \quad \exists\delta.\ \langle p,\ \delta,\ p\rangle \in \mathfrak{R}$$

i.e. we take the restriction of reflexive correspondences on $\mathcal{H}(p)|\mathfrak{R}$.

Instead of taking a specific process in the concerned equivalence class (cf. 3.10), one can use more elaborate construction which is more deterministic and hence which would be more satisfactory. Here we are content with the intuitive and tractable treatment in the above, since the following shows that not much is lost:

3.14 Proposition. *The above quotient defines a process structure. Moreover if we have two quotients of* $\mathbf{P}$ *with respect to* $\mathfrak{R}$*, then they are isomorphic.*

PROOF: We first show that $\mathcal{S}([p]_{\mathfrak{R}})$ forms a group. We note that any $\langle p,\ \delta,\ p\rangle \in \mathfrak{R}$ should include permutation of $\mathcal{H}(p)|\mathfrak{R}$ (since $\delta^{-1} \in \mathfrak{R}$ this should be the case). Then $id_p \in \mathfrak{R}$ since $\mathfrak{R}$ is reflexive, so the identity exists, the inverse always exists since $\mathfrak{R}$ is symmetric, and finally if two permutations are in $\mathcal{S}([p]_{\mathfrak{R}})$, then their composition is also in it by transitivity of $\mathfrak{R}$. Hence it forms a group.

Suppose there are two quotients from $\mathfrak{R}$. Then construct a p-map so that, for each $[p]_{\mathfrak{R}}$, there are correspondences such that: if the first quotient takes $\mathcal{H}(q_1)|\mathfrak{R}$ for $\mathcal{H}([p]_{\mathfrak{R}})$ and the second $\mathcal{H}(q_2)|\mathfrak{R}$ for the same, then $\langle [p]_{\mathfrak{R}},\ \delta,\ [p]_{\mathfrak{R}}\rangle$ if and only if $q_1\mathfrak{R}_\delta q_2$ with $\mathtt{im}(\delta) = \mathcal{H}(q_2)|\mathfrak{R}$. We leave it to the reader to check such a map is really symmetry preserving. ■

Now we show how p-maps relate to this quotient construction.

3.15 Proposition.

(i) *Given an equivalence* $\mathfrak{R}$ *over* $\mathbf{P}$*, a* natural map $\eta_{\mathfrak{R}}$ *is the set of correspondences from* $\mathbf{P}$ *to* $\mathbf{P}/\mathfrak{R}$ *such that, for* $q \in [p]_{\mathfrak{R}}$*, we have a correspondence* $\langle q,\ \delta,\ [p]_{\mathfrak{R}}\rangle$ *if and only if* $q\mathfrak{R}_\delta p$ *with* $\mathtt{im}(\delta) = \mathcal{H}(p)|\mathfrak{R}$*, assuming* $\mathcal{H}(p)|\mathfrak{R}$ *is used for* $\mathcal{H}([p]_{\mathfrak{R}})$*. Then* $\eta_{\mathfrak{R}}$ *is a p-map surjective on processes.*

(ii) *Suppose a p-map* $\mathfrak{R} : \mathbf{P} \to \mathbf{Q}$ *covers all processes in* $\mathbf{Q}$*. Define a set of correspondences* $\sim'$ *over* $\mathbf{P}$ *by setting* $p \sim'_\delta p'$ *if and only if, for some* q*,* π_1 *and* π_2*, we have:*

$$p\mathfrak{R}_{\pi_1}q \ \wedge\ p'\mathfrak{R}_{\pi_2}q \ \wedge\ \delta = \pi_2^{-1} \circ \pi_1 \ .$$

Then $\sim = \sim' \cup \mathbf{ID_P}$ *is an equivalence p-relation. Moreover, there is an isomorphism* $\mathfrak{R}'$ *from* $\mathbf{P}/\sim$ *to* $\mathbf{Q}$ *such that* $\mathfrak{R} = \mathfrak{R}' \circ \eta_\sim$*.*

PROOF:

(i) We only have to show $\eta_{\mathfrak{R}}$ indeed defines a p-map, since surjectivity (on processes) is obvious from the definition. First, that $\eta_{\mathfrak{R}}$ really gives a p-relation is not so difficult: since $\mathfrak{R}$ itself is a p-relation, whenever $\langle q,\ \delta,\ p\rangle \in \mathfrak{R}$ with $\mathtt{im}(\delta) = \mathcal{H}(p)|\mathfrak{R}$ and ρ_q and ρ_p are some symmetries, we always have $\langle q,\ \rho_q \circ \delta \circ \rho_p,\ p\rangle\mathfrak{R}$ and $\mathtt{im}(\rho_q \circ \delta \circ \rho_p) = \mathcal{H}(p)|\mathfrak{R}$ again. The essence lies in uniqueness, since originally diverse correspondences can be found in $\mathfrak{R}$. This becomes possible due to the definition of symmetry for $[p]_{\mathfrak{R}}$. Suppose $\langle q,\ \delta,\ p\rangle\mathfrak{R}$ and $\langle q,\ \delta',\ p\rangle\mathfrak{R}$ with both having images in $\mathcal{H}(p)|\mathfrak{R}$. Then we can write

$$\delta' = (\delta \circ \delta^{-1}) \circ \delta'$$

But, by definition, $\delta \circ \delta^{-1} \in \mathcal{S}([p]_{\mathfrak{R}})$, hence done.

(ii) That $\sim'$ (hence $\sim$) is a p-relation and that $\sim$ is an equivalence are both easy. To show that there is an isomorphism $\mathfrak{R}'$, we should show three things: (1) there is a bijection at the level of processes, (2) it induces bijection at the level of handles, and (3) it gives the isomorphism of symmetries as specified in Proposition 3.9. Constructing a map from $\mathbf{P}/\sim$ to $\mathbf{Q}$ in the way so that $[p]_\sim$ is mapped to q by δ iff $p\mathfrak{R}_\delta q$ (remember $[p]_\sim$ is essentially $\mathcal{H}(p)|\mathfrak{R}$), (1) and (2) are not difficult. The essential point is (3). By symmetry preservation, we already know that, if $p\mathfrak{R}_\pi q$, the symmetries in processes of $[p]_\sim$ can be already found in q. On the other hand, let us assume $s \in \mathcal{S}(q)$. If $p\mathfrak{R}_\pi q$, then $p\mathfrak{R}_{pios} q$, hence $p \sim_{\pi^{-1} oso \pi} p$. In this way all symmetries are reflected conjugately in $\sim$. Thus (3) is shown and we are done. ■

Note that the above shows how both p-map and quotient drop some information (by cutting off handles and adding symmetries) and, thus, raise the abstraction level.

Having been through the basic constructions for process structures, it is time to clarify how the theory in this Chapter corresponds to the one in Section 2. Let us first go back to Proposition 2.20, where we proved a symmetry presentation of a rooted process structure maintains all essential information of the original structure. But we already know such a presentation is nothing but a process structure itself!

3.16 Proposition. *Suppose α is a symmetry presentation of some rooted process structure, i.e. a family $\{sym(P_i)\}_{i\in I}$ where I being the equivalence classes by $\overset{\mathrm{N}}{\sim}$ and $P_i \in i$ for each i. Then this defines a process structure with the following data:*

(i) Processes are given by the indices $\{i\}$.

(ii) Handles are given by $\mathcal{H}(i) = \mathcal{FN}(P_i)$.

(iii) Symmetries are given by $\mathcal{S}(i) = sym(P_i)$.

Moreover we have $\alpha \leftrightarrow \beta$ if and only if $\alpha \simeq \beta$.

PROOF: That α is a process structure is immediate. For isomorphisms, clearly if there is an isomorphism in the sense of process structure, then they are isomorphic as symmetry presentations. Conversely, Proposition 3.9 (iii) shows that isomorphism between symmetry presentations is enough for generating an isomorphism as a p-map, hence done. ■

Thus, given any rooted process structure, if we take any one of its symmetry presentations, it is already a corresponding process structure. Moreover, from isomorphic *rooted* process structures, we always get isomorphic *non-rooted* process structures, by Proposition 2.20 and above.

As expected, there is also a way which does the converse. This needs some construction, but it also enjoys a virtue that, for any process structure, we can always get a canonical rooted version. We fix some $\mathcal{N}$ as always.

3.17 Proposition. *Given a process structure* $\mathbf{P}$*, we form an algebra by the following data:*

(i) The set of all injections from $\mathcal{H}(p)$ to $\mathcal{N}$ for each $p \in \mathbf{P}$, each written $\langle \Psi,\ p \rangle$ etc. and collectively as $\{\langle \Psi,\ p\rangle\}$.

(ii) Operations $[\sigma]$ defined as: $\langle \Psi,\ p\rangle[\rho] \stackrel{\text{def}}{=} \langle \rho \circ \Psi,\ p\rangle$ (well defined since $\rho \circ \Psi$ is again an injection from $\mathcal{H}(p)$ to $\mathcal{N}$), and a function $\mathcal{FN}(\cdot)$ defined as: $\mathcal{FN}(\langle \Psi,\ p\rangle) \stackrel{\text{def}}{=} \mathtt{im}(\Psi)$.

(iii) Relation $\approx$ over $\{\langle \Psi,\ p\rangle\}$ defined as:

$$\langle \Psi,\ p\rangle \approx \langle \Psi',\ p\rangle \quad \Leftrightarrow \quad \Psi' = \Psi \circ s \ \text{ for some } s \in \mathcal{S}(p).$$

Then $\approx$ *is a congruence over* $\{\langle \Psi,\ p\rangle\}$*, and* $\{\langle \Psi,\ p\rangle\}/\approx$ *defines a rooted process structure inheriting* $\mathcal{FN}$ *and* $[\rho]$ *from the underlying algebra. Moreover, writing* $[\![\mathbf{P}]\!]$ *for the resulting rooted process structure, we have, for any* $\mathbf{P}$ *and* $\mathbf{Q}$*,* $\mathbf{P} \simeq \mathbf{Q}$ *if and only if* $[\![\mathbf{P}]\!] \simeq [\![\mathbf{Q}]\!]$.

Proof: If $\langle \Psi,\ p\rangle \approx \langle \Psi',\ p\rangle$, i.e. $\Psi' = \Psi \circ s$ for some $s \in \mathcal{S}(p)$, then

$$\langle \Psi,\ p\rangle[\rho] \ = \ \langle \rho \circ \Psi,\ p\rangle \ \approx \ \langle \rho \circ \Psi \circ s,\ p\rangle \ = \ \langle \Psi',\ p\rangle[\rho]$$

and

$$\mathcal{FN}(\langle \Psi \circ s,\ p\rangle) \ = \ \mathtt{im}(\Psi \circ s) \ = \ \mathtt{im}(\Psi) \ = \ \mathcal{FN}(\langle \Psi,\ p\rangle)$$

So $[\sigma]$ and $\mathcal{FN}(\cdot)$ are well-defined over $\{\langle \Psi,\ p\rangle\}/\approx$. Moving to equations of Definition 2.1, associativity of $[\sigma]$ holds easily, while if ρ does not change names in $\mathtt{im}(\Psi)$, then $\sigma \circ \Psi = \Psi$, hence the second equation holds. Finally $\mathcal{FN}(\langle \Psi,\ p\rangle[\sigma]) = \mathtt{im}(\sigma \circ \Psi) = \sigma\mathcal{FN}(\langle \Psi,\ p\rangle)$, hence the third equation holds, so that we know it forms a rooted process structure. For isomorphisms, check that, for any $p \in \mathbf{P}$, the set of all rooted processes of the form $[\langle \Psi,\ p\rangle]_\approx$ is an orbit and, for any process in this orbit, say P, $sym(P)$ and $\mathcal{S}(p)$ are conjugately related by some bijection between $\mathcal{FN}(P)$ and $\mathcal{H}(p)$. Using these data, construct an isomorphism in one domain from the one in another domain in each turn, whose exact construction we leave to the reader. ■

Let us assume a function $\langle\!\langle \cdot \rangle\!\rangle$ which, to each rooted process structure, gives one of its symmetry presentations definitely. Whatever the choice we may take for $\langle\!\langle \cdot \rangle\!\rangle$, $[\![\cdot]\!]$ and $\langle\!\langle \cdot \rangle\!\rangle$ are inverse to each other, in the following sense:

3.18 Proposition.

(i) *For any rooted process structure* A*, we have* $[\![\langle\!\langle A \rangle\!\rangle]\!] \simeq A$.

(ii) *For any process structure* $\mathbf{P}$*, we have* $\langle\!\langle [\![\mathbf{P}]\!] \rangle\!\rangle \simeq \mathbf{P}$.

PROOF: We prove only (ii), since, by Propositions 3.16 and 3.17, we know (i) and (ii) are equivalent. Given process structures $\mathbf{P}$ and $\langle\!\langle [\![\mathbf{P}]\!] \rangle\!\rangle$, we construct a bijection between processes which induce a conjugate isomorphism for each pair, then use Proposition 3.9 (iii). But this is easy since, first, by $[\![\cdot]\!]$, each $p \in \mathbf{P}$ is mapped to a single orbit which includes $P = [\langle \Psi,\ p \rangle]_{\approx}$, for which $\mathcal{S}(p)$ and $sym(P)$ are conjugate by Ψ, then by $\langle\!\langle \cdot \rangle\!\rangle$ this is mapped, essentially speaking, to $sym(P)$ itself, so that we already have the required map. This concludes the proof. ■

Since we have accumulated enough knowledge about how each object in one form can be mapped to the one in another without losing essential information, our next task is to show there are similar exact correspondence in their ambient universes. In rooted process structures, our choice was a natural one, RPS_{rel} for relations and RPS for homomorphisms. For process structures, the only possibility would be to take p-relations and p-maps.

3.19 Proposition. *The set of process structures and p-relations among them forms a complete boolean allegory and includes, as its subcategory, the category of process structures and p-maps. The former is called* PS_{rel}*, while the latter is called* PS*.*

We devote the rest of the Chapter to the outline of the proof of:

3.20 Theorem. RPS_{rel} *and* PS_{rel} *are equivalent as allegories, where involved functors induce categorical equivalence between* RPS *and* PS*.*

3.21 Definition. (Relational Mappings)

(i) Given $\mathcal{R} : A \to A'$, we define $\mathcal{R}^\circ$ as the minimum set of correspondences from $\langle\!\langle A \rangle\!\rangle$ to $\langle\!\langle A' \rangle\!\rangle$ satisfying:

$$\langle P,\ Q \rangle \in \mathcal{R} \quad \Rightarrow \quad \langle p,\ \pi \circ \rho^{-1},\ q \rangle \in \mathcal{R}^\circ \ \text{ where } p \in sym(P[\rho]) \text{ and } q \in sym(Q[\pi]).$$

Above we are assuming that, from P's orbit, $\langle\!\langle \cdot \rangle\!\rangle$ selects $sym(P[\rho])$ as a symmetry presentation.

(ii) Given $\mathfrak{R} : \mathbf{P}_1 \to \mathbf{P}_2$, we define $\mathfrak{R}^\star$ as the minimum relation from $[\![\mathbf{P}_1]\!]$ to $[\![\mathbf{P}_2]\!]$ satisfying:

$$\langle p,\ \delta,\ q \rangle \in \mathfrak{R} \quad \Rightarrow \quad \langle [\langle \Psi_1,\ p \rangle]_{\approx},\ [\langle \Psi_2,\ q \rangle]_{\approx} \rangle \in \mathfrak{R}^\star$$

such that $\delta = {\Psi'_2}^{-1} \circ {\Psi_1}'$ with $\Psi'_i \approx \Psi_i$, $i = 1, 2$.

3.22 Proposition.

(i) *If $\mathcal{R}$ is a compatible relation, then $\mathcal{R}^\circ$ is a p-relation. In particular:*

(1) $\emptyset^\circ = \emptyset$.

(2) *If $\mathcal{R}$ is the compatible closure of a single tuple, then $\mathcal{R}^\circ$ is an abstract correspondence (cf. 3.6).*

(3) $(\bigcup_i \mathcal{R}_i)^\circ = \bigcup_i(\mathcal{R}_i^\circ)$, $(\bigcap_i \mathcal{R}_i)^\circ = \bigcap_i(\mathcal{R}_i^\circ)$, $(\mathcal{R}^C)^\circ = (\mathcal{R}^\circ)^C$, $(\mathcal{R}^{-1})^\circ = (\mathcal{R}^\circ)^{-1}$, as well as $\mathcal{R}_2 \circ \mathcal{R}_1^\circ = \mathcal{R}_2^\circ \circ \mathcal{R}_1^\circ$.

(ii) *If $\mathfrak{R}$ is a p-relation, then $\mathfrak{R}^\star$ is a compatible relation. In particular:*

(1) $\emptyset^\star = \emptyset$.

(2) *If $\mathfrak{R}$ is an abstract correspondence, then $\mathfrak{R}^\star$ is the compatible closure of a single tuple.*

(3) $(\bigcup_i \mathfrak{R}_i)^\star = \bigcup_i(\mathfrak{R}_i^\star)$, $(\bigcap_i \mathfrak{R}_i)^\star = \bigcap_i(\mathfrak{R}_i^\star)$, $(\mathfrak{R}^C)^\star = (\mathfrak{R}^\star)^C$, $(\mathfrak{R}^{-1})^\star = (\mathfrak{R}^\star)^{-1}$, as well as $\mathfrak{R}_2 \circ \mathfrak{R}_1^\star = \mathfrak{R}_2^\star \circ \mathfrak{R}_1^\star$.

(iii) *For a compatible relation $\mathcal{R}$, we have $\mathcal{F}_1 \circ \mathcal{R}^{\circ\star} \circ \mathcal{F}_2 = \mathcal{R}$ where $\mathcal{F}_{1,2}$ are isomorphisms. Similarly for a p-relation $\mathfrak{R}$, we have $\mathfrak{R}_1 \circ \mathfrak{R}^{\star\circ} \circ \mathfrak{R}_2 = \mathfrak{R}$ where $\mathfrak{R}_{1,2}$ are isomorphisms.*

Proof: For (i), we should show that the resulting set of correspondences is closed under $\sim$. It is easy to know that, without loss of generality, we can set:

$$\delta = \langle sym(P[\rho]),\ \pi \circ \rho^{-1},\ sym(Q[\pi])\rangle \in \mathcal{R}^\circ$$

which comes from $\langle P,\ Q\rangle \in [\![R]\!]$. Now if $\rho' \in sym(P[\rho])$, by compatibility and observing $\rho'^{-1} \in sym(P[\rho])$ too, we have $\langle P[\rho][\rho'^{-1}],\ Q[\rho][\rho'^{-1}]\rangle \in \mathcal{R}$ so that, finally, $\langle P[\rho],\ Q[\pi][pi^{-1}][\rho][\rho']\rangle \in \mathcal{R}$ from which (by definition) we know $\langle p,\ \pi\circ\rho^{-1}\circ\rho',\ q\rangle \in \mathcal{R}^\circ$. Similarly for the case when we take a symmetry of $Q[\pi]$.

Next we should show that if $\mathcal{R} = \{\langle P[\rho],\ Q[\rho]\rangle\}$ for some P and Q, then $\mathcal{R}^\circ$ is an abstract correspondence. From the above discussions, and due to the form of the mapping, we only have to consider the case for $\langle sym(P),\ \delta,\ sym(Q)\rangle$, which comes from $\langle P,\ Q\rangle \in \mathcal{R}$. But we can easily see that the correspondences from $sym(P)$ to $sym(Q)$ are all generated from tuples of the form $\langle P[\rho],\ Q[\sigma]\rangle$ for some $\pi \in sym(P)$ and $\rho \in sym(Q)$, and in these cases what we have is $\langle p,\ \pi^{-1} \circ \delta \circ \rho,\ q\rangle$, which is essentially equivalent to the original correspondence, hence done.

Finally the compatibility of operations with the map is mechanical.

For (ii), the only non-trivial reasoning is whether the resulting relation is compatible or not. Suppose, w.l.o.g., $\langle[\langle\Psi_1,\ p\rangle]_\approx,\ [\langle\Psi_2,\ q\rangle]_\approx\rangle \in \mathfrak{R}^\star$, which comes from $\langle p,\ \delta,\ q\rangle$, so that we have $\delta = \Psi_2 \circ \Psi_1$. But then for any σ we have $\delta = \Psi_2 \circ \rho^{-1} \circ \rho \circ \Psi_1$, so that we know $\langle[\langle\rho \circ \Psi_1,\ p\rangle]_\approx,\ [\langle\rho \circ \Psi_2,\ q\rangle]_\approx\rangle \in \mathfrak{R}^\star$, as required.

Now the above reasoning and the proof of 3.18, as well as the construction of relational mappings themselves, show how we can compose mappings to form an isomorphism from $\mathcal{R}$ to $\mathcal{R}^{\circ\star}$ and the one from $\mathfrak{R}$ to $\mathfrak{R}^{\star\circ}$. We leave the concrete construction of the isomorphisms to the reader. ∎

While this suffices for "allegorical" correspondences (take the isomorphisms in (iii) above as natural isomorphisms for the equivalence), we should also show the following:

3.23 Proposition.

(i) *If $\mathcal{R}$ is a homomorphism (of rooted process structures), then $\mathcal{R}^{\circ}$ is a p-map.*

(ii) *If $\mathfrak{R}$ is a p-map (of process structures), then $\mathfrak{R}^{\star}$ is a homomorphism.*

PROOF: For (i), suppose $\mathcal{R}$ is a homomorphism. We should derive three conditions for p-maps, i.e. uniqueness, surjectivity, and symmetry preservation, from the property of $\mathcal{R}$. Now we can easily verify that uniqueness (up to $\sim$) is derived from Proposition 2.11 (which says that if one process in an orbit in the source is mapped to an orbit in the target, then whatever process in the former orbit is always mapped to the same orbit), and surjectivity of correspondences is a direct consequence of Proposition 2.9, and symmetry preservation also directly follows from Proposition 2.18, hence done. For (ii), what we should know is that how three properties we just established can yield the property of being a function, i.e. uniqueness and everywhere-definedness, since compatibility is not a question any longer. But everywhere-definedness just comes from surjectivity of p-maps (on processes), so that the uniqueness as functions is the only remaining concern. The essential fact is that we need *both* uniqueness as p-maps *and* symmetry preservation to ensure the property in the resulting relation, used one by one. Firstly, the uniqueness in the sense of p-maps lets us have a constraint in the resulting compatible relation so that, if $\mathfrak{R}$ is a p-map, and $P\mathfrak{R}^{\star}Q$ and $P\mathfrak{R}^{\star}Q'$ holds, then $Q' = Q[\rho]$ should hold for some $\rho \in sym(P)$. So we have uniqueness up to symmetries of P. Next, the symmetry-preservation property of a p-map lets us have, whenever $P\mathfrak{R}^{\star}Q$ holds, for any $\rho \in sym(P)$, we have $\rho \restriction \mathcal{FN}(Q) \in sym(Q)$. Therefore we know that "Q may vary up to the symmetry of P" just means "Q may vary up to the symmetry of Q," that is, if P is decided Q is decided too (i.e. $Q = Q'$ in the above). Note, in this way, we can observe two clearly separated elements in homomorphisms, even within the realm of rooted process structures. Thus $\mathfrak{R}^{\star}$ is not only compatible but also everywhere-defined and single-valued, which, by definition, says that it is a homomorphism, hence done. ∎

This establishes Theorem 3.20 and answers to the question we posed at the end of the preceding Chapter.

References

[1] Abramsky, S., Interaction Categories, *Theory and Formal Methods 1993*, Springer-Verlag, 1993.

[2] Berry, G. and Boudol, G., The Chemical Abstract Machine. *Theoretical Computer Science*, vol 96, pp. 217–248, 1992.

[3] Cohn, P.M., *Universal Algebra*, D.Reidel, 1981.

[4] Freyd, P.J. and Scedrov, A., *Categories and Allegories*, North-Holland, 1990.

[5] Girard, J.-Y., Linear Logic, *Theoretical Computer Science*, Vol. 50, pp.1–102, 1987.

[6] Hoare, C. A. R., *Communicating Sequential Processes*, Prentice Hall, 1985.

[7] Honda, K. and Yoshida, N., On Reduction-Based Process Semantics, *13th. FST/TCS*, LNCS 761, pp.371–387, Springer-Verlag, 1993. The full version to appear in *Theoretical Computer Science.*

[8] Honda, K. and Yoshida, N., Combinatory Representation of Mobile Processes, *POPL'94*, pp.348–360, ACM press, 1994.

[9] Honda, K. and Yoshida, N., Replication in Concurrent Combinators, *TACS'94*, LNCS 789, pp.786–805, Springer-Verlag, 1994.

[10] Lafont, Y., Interaction Nets, *POPL'90*, pp. 95–108, ACM press, 1990.

[11] Meseguer, J., Conditional Rewriting Logic as a Unified Model of Concurrency, *Theoretical Computer Science*, vol 92, 1992.

[12] Milner, R., *A Calculus of Communicating Systems*, LNCS 76, Springer-Verlag, 1980.

[13] Milner, R., *Communication and Concurrency*, Prentice Hall, 1989.

[14] Milner, R., Functions as Processes. *Mathematical Structure in Computer Science*, 2(2), pp.119–146, 1992.

[15] Milner, R., *Action Structures.* Research Report LFCS-92-249, Computer Science Department, Edinburgh University, 1992.

[16] Milner, R., Parrow, J.G. and Walker, D.J., A Calculus of Mobile Processes, *Information and Computation* 100(1), pp.1–77, 1992.

[17] Yoshida, N., Graph Notation for Concurrent Combinators, to appear in *Proc.TPPP'94*, LNCS, Spinger-Verlag, 1995.