Convenient Category of Processes and Simulations I: Modulo Strong Bisimilarity

Duško Pavlović*

Abstract

Deep categorical analyses of various aspects of concurrency have been developed, but a uniform categorical treatment of the very first concepts seems to be hindered by the fact that the existing representations of processes as bisimilarity classes do not provide a sufficient account of computational morphisms.

In the present paper, we describe a category of processes modulo strong bisimulations, with the bisimilarity preserving simulations as morphisms, and show that it is isomorphic to — and can be conveniently represented by — a subcategory of transition systems, with graph morphisms. The representative of each process and every morphism can effectively calculated, using coinduction (but with no reference to proper classes). The method is applicable to richer notions of a process as well, which are studied in the sequel.

1 Introduction

A process is usually presented as some kind of a labelled graph. In fact, any directed graph with labelled edges and a distinguished initial node can be construed as an automaton [11, sec. 2.3]. The nodes are the states, the edges — the transitions. A computation, or a run, is a directed path, starting from the initial node. The string of labels, traced by a run, represents the accepted input data.

But of course, a graph is not completely determined by the paths through it. Many geometric properties of an automaton are not reflected in its computational behaviour. The presentation of processes by automata thus contains redundancies. To eliminate them, we quotient automata modulo computational equivalences, captured by various notions of *bisimulation*. Processes become equivalence classes of bisimilar automata, meant to be computationally indistinguishable.

However, working with large equivalence classes of automata tends to be cumbersome. One therefore seeks ways to uniformly choose a representative for each class. P. Aczel has proposed a method to obtain canonical representatives as elements of a coinductively defined proper class of non-wellfounded sets [2]. In [4]. M. Barr has explained how non-wellfounded sets can in be avoided in this construction. In section 4, we further modify Aczel's construction, as to be able to calculate the canonical representative of a given process effectively, i.e. within an arbitrary complete topos, and with no reference to proper classes. Of course, we only calculate one representative at a time, rather than the class of all of them at once.

The main point of it all, though, is not so much the effectiveness, as the need for incorporating the computational morphisms in the picture of processes[1]. The existing representation theories provide very little information about the process morphisms. Morphisms sometimes do appear in the literature on concurrency [8, 12], but in principle just to help capturing the bisimilarity. Morphisms *between* the bisimilarity classes have not yet been treated, it seems. A categorical analysis of the universal properties of the operations arising in process calculi is herewith blocked off before it could properly start. Without morphisms, it is

* Deptepartment of Computing, Imperial College of Science, Technology and Medicine, London SW7 2BZ, England; e-mail: D.Pavlovic@doc.ic.ac.uk

[1] Indeed, while defining an object as an element of a proper class does not seem very practical, displaying it as an object of a such-and-such category, albeit large, may suffice for a structural analysis.

hard to say clearly *what is the smallest process able to simulate those given processes.* A convenient category of processes should, of course, display it as their coproduct. This coproduct will then probably turn out to be the “objective” aspect of a familiar construct; but the categorical view may also disclose some less intuitive, yet computationally quite meaningful structures.

Having summarized the basic concepts of automata, simulations etc. in the first two parts of section 2, we take up formalizing the notion of a category of processes in subsection 2.3. This determines the target for representation theory, but also brings to attention some fairly conceptual, yet apparently neglected conditions on simulations, which tie them together in *uniform* and *bisimilarity preserving* families. Such families constitute the process morphisms.

Technically, our representation of the category of processes is built upon a characterisation of simulations in terms of bisimulation morphisms, presented in section 3. It can be viewed as a combination of the ideas of Castellani [8] with the categorical analysis due to Joyal, Nielsen and Winskel [12]. The actual representation is described in section 4. A process is represented by the smallest bisimulation quotient of any of its representatives. The effective calculation of such representatives, obtained by “localizing” the Aczel-style coinduction under a given transition system, is explained in subsection 4.3. As a consequence of the representation theorem, 4.5, the category of processes appears as a reflective subcategory of the category of reachable automata with the bisimilarity preserving graph morpisms. Other interesting subcategories, representing richer notions of a process, will be described in the sequel to this paper.

2 From automata to processes

In this section, we reformulate some familiar definitions categorically. Any sufficiently rich category will accomodate them: in principle, a topos with limits will suffice for all constructions of this paper. The reader is thus welcome to assume her/his favourite model of set theory.

2.1 Labelled graphs

Let Σ be a fixed set. A (Σ-)automaton P is a diagram

$$\Sigma \xleftarrow{\lambda} T_P \underset{\varrho}{\overset{\delta}{\rightrightarrows}} S_P \xleftarrow{\iota} 1 \tag{1}$$

in some “category of sets” $\mathcal{S}$. The operations δ, ϱ and λ assign to each *transition* from T_P respectively a *source* and a *target state* in S_P, and a *label* from the *alphabet* Σ. The constant ι is the *initial* state.

Automata morphisms are just the graph morphisms that preserve the labels and the initial state. A morphism $\varphi : P \to Q$ is thus a pair of functions $\langle {}^T\varphi, {}^S\varphi \rangle$, making the diagram

$$\begin{array}{ccccccc} & \overset{\lambda}{\swarrow} & T_P & \underset{\varrho}{\overset{\delta}{\rightrightarrows}} & S_P & \overset{\iota}{\nwarrow} & \\ \Sigma & & \downarrow {}^T\varphi & & \downarrow {}^S\varphi & & 1 \\ & \underset{\lambda}{\nwarrow} & T_Q & \underset{\varrho}{\overset{\delta}{\rightrightarrows}} & S_Q & \underset{\iota}{\swarrow} & \end{array} \tag{2}$$

commute in the obvious way. Objects (1) and morphisms (2) form the category $\mathcal{A}$ of automata.

A *transition system* is an automaton where the function $\langle \delta, \lambda, \varrho \rangle : T \longrightarrow S \times \Sigma \times S$ is injective. This means that there is at most one transition from x to x' with the label a, which can thus be unequivocally denoted $x \stackrel{a}{\to} x'$. Having several such transitions clearly changes nothing in the computational behaviour of the automaton. In concurrency, one usually restricts considerations to transition systems. Although it involve one additional condition on diagrams (1), this restriction often simplifies arguments. Transition systems span a reflective subcategory $\mathcal{A}^t$ in $\mathcal{A}$: an automaton is reflected by taking the image of the function $\langle \delta, \lambda, \varrho \rangle$, i.e. identifying any two transitions with the same source, target and label. Most results about $\mathcal{A}$ are carried over to $\mathcal{A}^t$ along this reflection.

On the other hand, one could restrict $\mathcal{A}$ to *reachable* automata, i.e. those where each state can be reached from ι by some run. In computation, the unreachable states play no role. The reachability requirement induces a coreflective subcategory $\mathcal{A}^r$ of $\mathcal{A}$: the coreflection simply deletes the unreachable states, together with all the transitions involving them. The intersection $\mathcal{A}^{rt}$ of $\mathcal{A}^r$ and $\mathcal{A}^t$ is reflective in $\mathcal{A}^r$, and coreflective in $\mathcal{A}^t$. The reflection and the coreflection commute.

$$\begin{array}{ccc} \mathcal{A} & \underset{\hookleftarrow}{\overset{\longrightarrow}{\bot}} & \mathcal{A}^t \\ \downarrow \vdash \uparrow & & \downarrow \vdash \uparrow \\ \mathcal{A}^r & \underset{\hookleftarrow}{\overset{\longrightarrow}{\bot}} & \mathcal{A}^{rt} \end{array} \tag{3}$$

Some of our results will require reachability; others are only valid for transition systems. In principle, we should thus work within the category $\mathcal{A}^{rt}$. It is, however, not completely uninformative to keep all four categories (3) in the game; and it often proves to be a technical advantage.

2.2 (Bi)simulations internally

All categories described so far are regular, and thus accomodate internal calculi of relations [9, sec. 1.5]. Simulations and bisimulations can thus be studied internally. But we first have to extend the standard definitions to transitions.

In any category, an internal relation R from P to Q is simply a subobject $R \hookrightarrow P \times Q$, conveniently written $P \leftarrow R \rightarrow Q$. In our setting, a relation $P \leftarrow R \rightarrow Q$ thus consists of two ordinary relations on sets, $S_P \leftarrow S_R \rightarrow S_Q$ and $T_P \leftarrow T_R \rightarrow T_Q$, such that any two T_R-related transitions must have S_R-related sources, S_R-related targets, and the same labels. Of course, the initial states must be S_R-related too.

To simplify notation, we work in $\mathcal{A}^t$, i.e. assume that there is at most one transition $x \overset{a}{\to} x'$. However, with more care, the same definitions go through for arbitrary automata.

Definition 2.1 *A* simulation *from P to Q is a relation $P \leftarrow R \rightarrow Q$ in $\mathcal{A}$ satisfying*

$$x \overset{a}{\to} x' \wedge x S_R y \implies \exists y'.\ y \overset{a}{\to} y' \wedge x' S_R y' \tag{4}$$

$$x S_R y \wedge x' S_R y' \implies \left(x \overset{a}{\to} x'\right) T_R \left(y \overset{a}{\to} y'\right) \tag{5}$$

If there is such a relation, we say that P is simulated *by Q.*

A bisimulation *is a simulation $P \leftarrow R \rightarrow Q$ such that the opposite relation $Q \leftarrow R^o \rightarrow P$ is a simulation too. Automata P and Q are said to be* bisimilar *if there is a bisimulation between them.*

The standard definition of a simulation only speaks of states. It requires that the initial states are related, and that (4) holds — i.e. that

$$\text{every span } \begin{array}{cc} x & \overset{a}{\longrightarrow} x' \\ \vdots R & \\ y & \end{array} \quad \text{can be completed to a square } \begin{array}{ccc} x & \overset{a}{\longrightarrow} & x' \\ \vdots R & & \vdots R \\ y & \overset{a}{\longrightarrow} & y' \end{array}. \tag{6}$$

Condition (5) extends this to transitions. Its converse, the requirement that any two related transitions must have related sources and related targets, says that the following square must commute.

$$\begin{array}{ccc} T_R & \xrightarrow{\langle \delta, \varrho \rangle} & S_R \times S_R \\ \downarrow & & \downarrow \\ T_P \times T_Q & \xrightarrow[\langle \delta\times\delta, \varrho\times\varrho \rangle]{} & S_P \times S_Q \times S_P \times S_Q \end{array} \tag{7}$$

So it must hold by virtue of internal relations. Condition (5), on the other hand, says that two transitions with the same label must be R-related as soon as their sources, as well as their targets, are related. This means that square (7) must be a pullback. Thus, the state component $S_P \leftarrow S_R \rightarrow S_Q$ completely determines the simulation $P \leftarrow R \rightarrow Q$. So we can write xRy instead of xS_Ry. The notions resulting from 2.1 are equivalent with the standard ones, just adapted for categories.

Automata and simulations form a subcategory of the category of relations on automata. It is, of course, poset-enriched with inclusions. A meet of two simulations usually does not exist, but the union of any inhabited set of simulations is always a simulation. Hence, if there is a simulation from P to Q, then there must exist the largest one. Obviously, all this holds for bisimulations too. The largest bisimulation from P to Q will be denoted $P \leftarrow\!\sim\!\rightarrow Q$.

2.3 Processes

The bicategory of processes should now be obtained by quotienting automata and simulations by bisimilarity. A processes should be an equivalence class of bisimilar automata. Let us denote them by Π, Θ. A process morphism $\Pi \leftarrow \Xi \rightarrow \Theta$ should be a uniform family Ξ of computationally equivalent simulations $P \leftarrow R \rightarrow Q$, one for each $P \in \Pi$ and $Q \in \Theta$.

The uniformity of Ξ could be captured as follows. Since any two elements P and P' of Π are identified along $P \leftarrow\!\sim\!\rightarrow P'$, and ditto in Θ, then any element $P \leftarrow R \rightarrow Q$ of Ξ could determine any other element $P' \leftarrow R' \rightarrow Q'$ by composing with $\sim$ on both sides:

$$x'R'y' \iff \exists xy.\ x \sim x' \wedge xRy \wedge y \sim y' \tag{8}$$

In this way, a process morphism would be determined by any of its elements, just like a process itself is determined by any representative. With this in mind, we impose condition (9) below. Condition (10), on the other hand, just says that a process morphism should preserve the bisimilarity.

In capturing these ideas, some trivial, yet tedious problems with unreachable states are encountered. To avoid them, we restrict processes to $\mathcal{A}^r$. In the sequel, Π, Θ *will invariably denote bisimilarity classes of reachable automata.*

Definition 2.2 *A* process morphism $\Pi \leftarrow \Xi \rightarrow \Theta$ *is a class of simulations*

$$\Xi = \{P \leftarrow R \rightarrow Q | P \in \Pi, Q \in \Theta\},$$

such that for any $R, R' \in \Xi$ *holds*

$$x \sim x' \wedge xRy \wedge y \sim y' \implies x'R'y' \tag{9}$$
$$x \sim x' \wedge xRy \wedge x'R'y' \implies y \sim y' \tag{10}$$

for all $x \in P, y \in Q, x' \in P', y' \in Q'$.

A simulation $P \leftarrow R \rightarrow Q$ *satisfying* (9) *with* $R' = R$ *is called* saturated. *If it satisfies* (10)*, again with* $R' = R$*, then it is* sober. *A* process simulation *is required to be both saturated and sober.*

In pictures, (9) says that the simulation obtained by going

$$\begin{array}{ccccc} \Pi & \quad & P & \leftharpoondown\!\sim\!\rightharpoondown & P' \\ \upharpoonleft & & \upharpoonleft & & \upharpoonleft \\ \Xi & & R & & R' \\ \downharpoonleft & & \downharpoonleft & & \downharpoonleft \\ \Theta & & Q & \leftharpoondown\!\sim\!\rightharpoondown & Q' \end{array} \tag{11}$$

from P' to Q' *via* P and Q is contained in $P' \leftarrow R' \rightarrow Q'$. On the other hand (10) says that going from Q to Q' *via* P and P' yields a relation contained in $Q \leftarrow\!\sim\!\rightarrow Q'$. Together, both conditions say that for all $x' \in P'$ and $y \in Q$ holds

$$\exists x \in P.\ x \sim x' \wedge xRy \quad \Longrightarrow \quad \forall y' \in Q'.\ y \sim y' \Leftrightarrow x'R'y' \tag{12}$$

The process morphisms are built from process simulations — i.e. sober and saturated. The unsober ones would blur the image of a process by *distinguishing what is computationally indistinguishable.* Conceptually and technically, the sobriety is a *sine qua non.* The saturation requirement is not: the bicategory of processes could be defined without it — but the price would be having a proper class of morphisms between any two processes. The saturation ensures that

Proposition 2.3 *Each process morphism is uniquely determined by any of its components.*

so that there are exactly as many process morphisms between Π and Θ as process simulations from any $P \in \Pi$ to any $Q \in \Theta$.

Another benefit from the saturation is that we are getting an ordinary category, rather than a bicategory of processes.

Lemma 2.4 *Each simulation $P \leftarrow R \rightarrow Q$ is included in a saturated one: the saturation $P \leftarrow R' \rightarrow Q$ can be defined as in* (8)*. If R is sober, the saturation R' is not only the smallest saturated, but also the largest sober simulation containing it. Hence, each sober simulation is contained in a unique process simulation.*

Proof. The only nontrivial part is showing that R' is the largest saturated simulation containing R.

Note, first of all, that a simulation R is sober if and only if it relates the elements of a $\sim$-equivalence class from the reachable part of P to the elements of a *unique* $\sim$-equivalence class from the reachable part of Q. A sober simulation R thus induces a mapping from the $\sim$-classes of P to the $\sim$-classes of Q.

The saturation R' now relates *every* pair of the elements from the corresponding $\sim$-equivalence classes. In particular, xRy and $y \sim z$ imply $xR'z$. Therefore, $\langle x, z\rangle \notin R'$ implies $z \not\sim y$, for all y such that xRy. This means that no proper extension of R' can be sober. □

The preceding proposition implies that any two comparable process simulations must coincide. Hence the *ordinary* category $\mathcal{P}$ of processes. Its abstract construction and the universal property are described in the appendix of [18]. It is a quotient of the bicategory $\ddot{\mathcal{A}}$ of reachable automata and sober simulations: the functor $\ddot{\mathcal{A}} \longrightarrow \mathcal{P}$ sends each automaton to the bisimilarity class of its reachable part, and each sober simulation to the process morphism induced by the saturation of its reachable part. In a formal sense, this functor is initial among all functors quotienting the bisimilarity. The representation developed in the sequel will show that it has a full and faithful right adjoint. On the other hand, proposition 2.3 implies that its restriction to sober *saturated* simulations is a weak equivalence! The task of representation theory boils down to providing a suitable right inverse for this equivalence. It will be constructed in such a way, that the process morphisms will be represented by certain automata morphisms, the graphics of which happen to be process simulations. The next section will lead up to such a representation.

3 (Bi)simulations intrinsically

Although they can be formulated in any regular category, conditions (4–5), defining a simulation, are not really suitable to work with in a category. We shall now provide more "categorical" characterisations. They summarize some familiar, often rediscovered ideas.

When P is a reachable automaton, every simulation $P \leftarrow R \rightarrow Q$ is clearly a total relation. This means, of course, that the projection $R \to P$ is an epimorphism. The idea is now to strengthen this totality assumption sufficiently to be able toprove the converse — that R is a simulation. It turns out that "$R \to P$ is an epimorphism" needs to be strengthened to "$R \to P$ is a bisimulation morphism". In a sense, simulations are the "bisimulation total" relations. This idea will be formalized in theorem 3.3.

Definition 3.1 *We say that $\varphi : P \to Q$ in $\mathcal{A}$ is a* bisimulation *(resp.* sober*) morphism if its graphic $P \stackrel{\mathrm{id}}{\leftarrow} P \stackrel{\varphi}{\to} Q$ is a bisimulation (resp. sober simulation).*

When $\varphi : P \to Q$ is a bisimulation morphism, we say that P is a *bisimulation cover* of Q and that Q is a *bisimulation quotient* of P. However, note that a bisimulation cover may not be actual epimorphism: it may not cover the unreachable states; and a bisimulation quotient is an acutal quotient only if Q is a reachable automaton.

The graphic of a morphism $\varphi : P \to Q$ is always a simulation. It is sober if moreover $x \sim x'$ implies $\varphi(x) \sim \varphi(x')$. Saturated morphisms could be defined in the same way, but they are less interesting: the graphic of $\varphi : P \to Q$ is saturated if and only if $x \sim x' \Rightarrow \varphi(x) \sim \varphi(x')$ holds for all $x, x' \in P$, and $y \sim y' \Rightarrow y = y'$ for all $y, y' \in Q$. Most automata thus preclude saturated morphisms. However, we shall soon extract a class on which *all* morphisms must be saturated.

Lemma 3.2 *Any morphism $\varphi : P \to Q$ of automata is bisimulation if and only if the square*

$$\begin{array}{ccc} T_P & \stackrel{\delta}{\longrightarrow} & S_P \\ {\scriptstyle {}^T\varphi}\downarrow & & \downarrow{\scriptstyle {}^S\varphi} \\ T_Q & \underset{\delta}{\longrightarrow} & S_Q \end{array} \tag{13}$$

is a quasi-pullback.

By definition, a square is a quasi-pullback if it factors through the actual pullback by an epimorphism. $\varphi : P \to Q$ is thus open if and only if the obvious arrow $T_P \longrightarrow T_Q \times_{S_Q} S_P$ is an epi.

Proof of 3.2. Instantiated to the relation $R = \langle \varphi, \mathrm{id} \rangle$, condition (4) says that for every $(x \stackrel{a}{\to} x') \in T_Q$ and for every $y \in S_P$ with $x = {}^S\varphi(y)$, there is some $(y \stackrel{a}{\to} y') \in T_P$ with $x' = {}^S\varphi(y')$ and $(x \stackrel{a}{\to} x') = {}^T\varphi(y \stackrel{a}{\to} y')$. But this just means that (13) is a quasi-pullback. □

Since a simulation from a reachable Q must be total, every quasi-pullback (13) corresponding to a morphism in $\mathcal{A}^r$ must in come from an epimorphism, and both ${}^S\varphi$ and ${}^T\varphi$ must be surjective. However, lemma 3.2 holds for arbitrary automata, as well as the following theorem.

Theorem 3.3 **(a)** *A relation $P \leftarrow R \to Q$ in $\mathcal{A}$ is a (sober) simulation if and only if R is a bisimulation cover of P (and $R \to Q$ is sober).*

(b) *The transition systems P and Q are bisimilar if and only if they have a common bisimulation cover; or equivalently* **(c)** *if and only if they have a common bisimulation quotient.*

Proof. Part **(a)** can be proved just like 3.2, by considering square (13) for $\varphi = p : R \to P$, and showing that (4) says that it is a weak pullback.

Alternatively, we could decompose the relation $P \stackrel{p}{\leftarrow} R \stackrel{q}{\to} Q$ as $R = qp^o$, where q denotes its own graphic

$R \xleftarrow{\text{id}} R \xrightarrow{q} Q$, while p^o is $P \xleftarrow{p} R \xrightarrow{\text{id}} R$. In this setting, statement (a) says that R is a simulation if and only if p^o is. Since q is always a simulation, (a) boils down to simple closure properties of simulations.

(As for the sobriety part, note that a bisimulation is always sober. Hence, when p is a bisimulation, R preserves the bisimilarity if and only if q does.)

Part **(b)** follows immediately from (a): we know that $P \leftarrow R \rightarrow Q$ is a simulation if and only if R covers P, and that $Q \leftarrow R^o \rightarrow P$ is a simulation if and only if R covers Q. So R is a bisimulation if and only if it covers both.

Towards a proof of **(c)**, first notice that 3.2 implies that bisimulation morphisms are pullback stable. Given a common bisimulation quotient, i.e. an opspan $P \xrightarrow{\alpha} M \xleftarrow{\beta} Q$ of bisimulation morphisms, a common bisimulation cover R can be obtained as a pullback $P \leftarrow R \rightarrow Q$ of α and β.

The other way around, a bisimulation $P \xleftarrow{p} R \xrightarrow{q} Q$ induces the common bisimulation quotient M as a pushout of p and q. Bisimulation morphisms are not stable under arbitrary pushouts, but they are stable under pushouts along bisimulation morphisms. To see why, we describe M in some detail[2]

Call *zigzag* a finite sequence $\langle z_0, z_1, \ldots z_n \rangle$ from $S_P + S_Q$, such that

$$z_0 R^\bullet z_1 R^\bullet z_2 \ldots z_{n-1} R^\bullet z_n,$$

where each $R^\bullet$ is either R or R^o. They alternate and each zigzag thus alternates between S_P and S_Q. Let $z\overline{R}z'$ denote that there is a zigzag from z to z', or $z = z'$. This is the equivalence relation generated by R.

The states of M are now the equivalence classes induced by $\overline{R}$. The transitions of M are the equivalence classes of transitions from P and Q, modulo the obvious extension of $\overline{R}$.

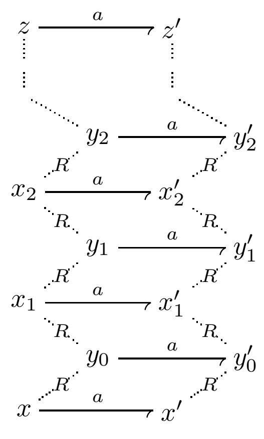
The morphisms $\alpha : P \rightarrow M$ and $\beta : Q \rightarrow M$ are induced by postcomposing the inclusions $S_P \hookrightarrow S_P + S_Q$ and $S_Q \hookrightarrow S_P + S_Q$ with the quotient map $S_P + S_Q \rightarrow S_M$. Using 3.2, we show that $\alpha : P \rightarrow M$ is a bisimulation whenever R is.

Given a state X in M and a transition $X \xrightarrow{a} X'$, for any state $x \in X$ (i.e. $\alpha(x) = X$), we must find a state $x' \in X'$ (i.e. $\alpha(x') = X'$), with $x \xrightarrow{a} x'$.

Let $z \xrightarrow{a} z'$ be an arbitrary representative of $X \xrightarrow{a} X'$. This implies $z \in X$. Since X is an $\overline{R}$-equivalence class, there is a zigzag

$$zR^\bullet \ldots y_2 R^o x_2 R y_1 R^o x_1 R y_0 R^o x \tag{14}$$

Since R is a bisimulation, we can follow this zigzag and find an a-transition from each element of (14) to some element of X'.



□

[2] A categorically minded reader should have no trouble transforming the following set-theoretical explanation into a formal argument over pretopos.

Remark. The bisimulation morphisms have been considered in various contexts, under a host of bisimilar names: p-morphisms [20], zig-zag morphisms [6], abstraction homomorphisms [8], pure morphisms [5], open maps [12]. I hope the reader will forgive my modest contribution to this terminological variety.

In [12], Joyal, Nielsen and Winskel have proposed a uniform categorical treatment for various concepts of bisimulation. It is based on characterisation 3.3(b), their theorem 5, plus the observation that various classes of bisimulation morphisms are generally characterized by a weak lifting property with respect to different classes of arrows. The bisimulation morphisms considered here are their Bran_Σ-open maps. On the other hand, characterisation 3.3(c) is related to Castellani's approach in [8].

As a consequence of the sobriety part of 3.3(a), we get

Corollary 3.4 *The sober simulations are just any simulations in the category of automata restricted to sober morphisms. The poset-enriched category sober simulations $\ddot{A}$ is isomorphic with the category of simulations in the category $\dot{A}$ of sober morphisms.*

4 Representing processes

According to 3.3, processes are just the connected components of the category $\mathcal{A}^r_\sim$ of reachable automata and the bisimulation morphisms. It turns out that each of these connected components has a terminal object: the smallest bisimulation quotient of all of its objects. This is, of course, a natural candidate for a representative.

4.1 Irredundant automata

In any category, the terminal object of a connected component appears as a *subterminal* object of the whole: it receives at most one arrow from any other object (cf. appendix A). Processes will be represented by the subterminal objects of $\mathcal{A}^r_\sim$.

Definition 4.1 *An automaton is said to be* irredundant *if it is reachable and there is at most one bisimulation morphism to it from any reachable automaton.*

Theorem 3.3 implies that a reachable automaton P will be irredundant if and only if its only self-bisimulation is the equality relation $P \overset{\text{id}}{\leftarrow} P \overset{\text{id}}{\rightarrow} P$. In other words, any two bisimilar states of P must be equal. Clearly, the same must hold for transitions (cf. (5)). An irredundant automaton will thus always be a transition system.

Lemma 4.2 *If P is irredundant, every morphism $\varphi : P \to Q$ must be sober; if Q is irredundant too, then every φ must be saturated as well. On the other hand, if P is reachable and Q irredundant, any sober simulation $P \overset{p}{\leftarrow} R \overset{q}{\rightarrow} Q$ must be the graphic a morphism, i.e. p must be an iso.*

Proof. The sobriety of φ is a consequence of its single-valuedness and the fact that P satisfies $x' \sim x'' \Rightarrow x' = x''$. The saturation follows from the analogous propery of Q.

Towards the second statent, we first show that $p : R \to P$ is a monic. Namely, since $p(z)Rq(z)$ holds by definition for every $z \in R$,

$$p(z') = p(z'') \implies p(z') \sim p(z'') \overset{\text{(s)}}{\implies} q(z') \sim q(z'') \overset{\text{(i)}}{\implies} q(z') = q(z'') \tag{15}$$

follows from sobriety (s) of R and irredundancy (i) of Q. Since $\langle p, q\rangle : R \hookrightarrow P \times Q$ is a monic, (15) implies that p is a monic. On the other hand, by 3.3(a), it is a bisimulation morphism, and hence an epi, since P is reachable. p is thus an isomorphism, and the span $P \leftarrow R \rightarrow Q$ represents[3] the graphic of q. □

[3] Recall that relations are defined as subobjects, i.e. the *isomorphism classes* of such spans.

Directly from the preceding lemma, we get the arrow part of the representation theorem, summarized in the next proposition. The one below it yields the object part.

Proposition 4.3 *The process simulations between irredundant automata correspond bijectively to the automata morphisms between them.*

Proposition 4.4 *Each process* Π *contains, up to isomorphism, exactly one irredundant automaton* $\widehat{\Pi} \in \Pi$.

Proof. The irredundant representative $\widehat{\Pi}$ can be obtained from any $P \in \Pi$, as its smallest (couniversal) bisimulation quotient. The construction and the proof that this is an irridundant automaton is deferred for the next two subsections.

The smallest bisimulation quotient $\widehat{\Pi} \in \Pi$ of $P \in \Pi$ will automatically be the smallest bisimulation quotient of any other $P' \in \Pi$. Namely, by 3.3(c), $\widehat{\Pi}$ and P' must have a common bisimulation quotient. The minimality assumption on $\widehat{\Pi}$ implies that it cannot have any proper quotients. So it must be a quotient of P' itself. □

The above propositions imply that the category of processes is equivalent with the category $\mathcal{I}$ spanned in $\mathcal{A}$ by irredundant automata, and actually isomorphic with its skeleton $\underline{\mathcal{I}}$ (the quotient where any two isomorphic objects of $\mathcal{I}$ are identified). Indeed, $\mathcal{I}$ is equivalent with its skeleton, since two irredundant automata can be isomorphic in one way at most. Hence

Theorem 4.5 $\mathcal{P} \;\cong\; \underline{\mathcal{I}}$

Proof. The functor $\mathcal{P} \to \underline{\mathcal{I}}$ should send each process Π to (the isomorphism class of) the irredundant representative $\widehat{\Pi}$, as in 4.4. A process morphism $\Pi \leftarrow \Xi \to \Theta$ is then represented by its component running from $\widehat{\Pi}$ to $\widehat{\Theta}$. According to proposition 4.3, this component must be an automata morphism.

The functor $\underline{\mathcal{I}} \to \mathcal{P}$ sends each irredundant automaton (i.e. its isomorphism class) to the class of reachable automata bisimilar with it. An automata morphism is mapped to the family of simulations induced by its graphic *via* formula (8). This graphic is a process simulation by 4.3, and the induced family is a process morphism. □

Remark. The difference between $\mathcal{I}$ and $\underline{\mathcal{I}}$ should not be taken too seriously. It lies entirely within the foundational tedium of indeterminacy of set-theoretical encodings: indistinguishable mathematical structures can always be presented as marginally different sets.

4.2 Couniversal bisimulation quotients

To complete the proof of proposition 4.4, and thus of theorem 4.5, we shall now show that every reachable automaton has an irredundant quotient. It will be effectively constructed. The construction is based on the following fact, which is a consequence of proposition A.2 in the appendix.

Proposition 4.6 *A morphism* $P \to Q$ *is a couniversal bisimulation quotient of the reachable automaton* P *if and only if* Q *is irredundant.*

An irredundant representative of a process Π can thus be constructed as the terminal object in the category of the bisimulation quotients of any $P \in \Pi$.

In order to reduce this task to coinductive methods, we restrict to transition systems. On the other hand, for simplicity, let us neglect the reachability requirement for a while.

4.3 Coalgebras

We first describe a convenient presentation of the category $\mathcal{A}^t_\sim$ of transition systems and bisimulation morphisms. The following category $\mathcal{C}$ will turn out to be strongly equivalent with it.

The objects of $\mathcal{C}$ are the diagrams

$$1 \xrightarrow{\iota} S \xrightarrow{\sigma} \wp(\Sigma \times S) \tag{16}$$

where $\wp$ is the power-set functor. A morphism $\varphi : S \to S'$ in $\mathcal{C}$ is just a function that preserves ι and σ,

$$\begin{array}{ccccc} & & S & \xrightarrow{\sigma} & \wp(\Sigma \times S) \\ & {}^{\iota}\nearrow & & & \\ 1 & & \Big\downarrow{\scriptstyle \varphi} & & \Big\downarrow{\scriptstyle (\Sigma\times\varphi)_!} \\ & {}_{\iota}\searrow & & & \\ & & S' & \xrightarrow[\sigma]{} & \wp(\Sigma \times S') \end{array} \tag{17}$$

where $(\Sigma \times \varphi)_!$ is the direct image induced by $\Sigma \times \varphi$.

Proposition 4.7 $\mathcal{A}^t_\sim \;\simeq\; \mathcal{C}$

Proof. The correspondence of the objects is straightforward: σ is just another view of a relation on $S \times \Sigma \times S$, which denotes Σ-labelled transitions on S:

$$\frac{\dfrac{S \to \wp(\Sigma \times S)}{S \times \Sigma \times S \to \Omega}}{T \rightarrowtail S \times \Sigma \times S} \tag{18}$$

(Ω is the set of truth values — classically, just $\{\top, \bot\}$). Along this correspondence, each $\mathcal{C}$-morphism $\varphi : S \to S'$ induces a morphism like on (2). Of course, such a morphism on transition systems is completely determined by its state component. The other way around, though, a function $\varphi : S \to S'$, coming from an $\mathcal{A}^t$-morphism, is not necessarily a $\mathcal{C}$-morphism: it may not strictly preserve σ. Indeed, an assignment of T-transitions to some T'-transitions only ensures

$$(\Sigma \times \varphi)_! \circ \sigma \quad \subseteq \quad \sigma \circ \varphi$$

The opposite inclusion holds — and makes the square on (17) commutative — if and only if φ is a bisimulation morphism. The $\mathcal{C}$-morphisms exactly correspond to the bisimulation morphisms. This fact can be checked directly, or derived, perhaps more informatively, from lemmas 3.2 and B.1. The latter says that the square

$$\begin{array}{ccc} T & \xrightarrow{\delta} & S \\ {\scriptstyle \tilde{\varphi}}\Big\downarrow & & \Big\downarrow{\scriptstyle \varphi} \\ T' & \xrightarrow[\delta]{} & S' \end{array} \tag{19}$$

is a quasipullback if and only if the second square from the left in

$$\begin{array}{ccccccccc} & & & & \overset{\sigma}{\frown} & & & & \\ S & \xrightarrow[\{\}]{} & \wp S & \xrightarrow[\delta^*]{} & \wp T & \rightarrowtail & \wp(S \times \Sigma \times S) & \xrightarrow[\pi_!]{} & \wp(\Sigma \times S) \\ {\scriptstyle \varphi}\Big\downarrow & & \Big\downarrow{\scriptstyle \varphi_!} & & \Big\downarrow{\scriptstyle \tilde{\varphi}_!} & & \Big\downarrow{\scriptstyle (\varphi\times\Sigma\times\varphi)_!} & & \Big\downarrow{\scriptstyle (\Sigma\times\varphi)_!} \\ S' & \xrightarrow{\{\}} & \wp S' & \xrightarrow{\delta^*} & \wp T' & \rightarrowtail & \wp(S' \times \Sigma \times S') & \xrightarrow{\pi_!} & \wp(\Sigma \times S') \\ & & & & \underset{\sigma}{\smile} & & & & \end{array} \tag{20}$$

commutes. This square, however, commutes if and only if all of diagram (20) commutes. $\square$

Remarks. The representation of $\mathcal{A}^t_\sim$ in the form $\mathcal{C}$ is based on Aczel's coalgebraic view of transition systems [2, ch. 8]. Omitting the initial states ι, one is indeed left with coalgebras[4] $S \to \wp(\Sigma \times S)$, i.e. the functions assigning to each state $x \in S$ the set of pairs $\langle a, x' \rangle$ such that there is a transition $x \xrightarrow{a} x'$.

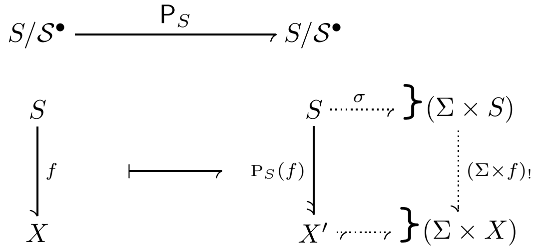
The main part of Aczel's theory is proving that there is a coinductively defined proper class $\mathcal{T} \cong \wp(\Sigma \times \mathcal{T})$, terminal for all coalgebras $S \to \wp(\Sigma \times S)$ — i.e., allowing a unique morphism from each of them — only too large to be a coalgebra itself. This turns out to be the class of canonical representatives of bisimilarity classes. Indeed, its elements correspond to its subcoalgebras, which are subterminal. The canonical representative of a transition system $S \to \wp(\Sigma \times S)$ thus appears as its image in $\mathcal{T} \to \wp(\Sigma \times \mathcal{T})$.

There are two points to be made here. The **first point** is that the relevant coalgebra constructions on transition systems go through virtually unchanged *with* the initial states. Although $\mathcal{C}$ is not an official category of coalgebras, the theory from [3, 4] carries over: carrying ι around is no problem.

The **second point** is that the excursion into proper classes is unnecessary: subterminal objects of $\mathcal{C}$ can be determined with no reference to a large "terminal object" on top of it. This is a simple technical remark, but it removes a contingent obstacle to *actual* calculation representatives! Indeed, an object defined by its membership in a proper class can hardly be regarded as effectively constructible.

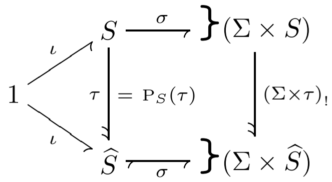
The construction. Let $1 \xrightarrow{\iota} S \xrightarrow{\sigma} \wp(\Sigma \times S)$ be a fixed transition system. We want to calculate its couniversal quotient in $\mathcal{C}$.

Denote by $\mathcal{S}^\bullet$ be the category of pointed sets, i.e. each with a distinguished element ι. The morphisms of $\mathcal{S}^\bullet$ are the ι-preserving functions. Let $S/\mathcal{S}^\bullet$ be the coslice of $\mathcal{S}^\bullet$ under $S \in \mathcal{S}^\bullet$. The objects of this category are thus the $\mathcal{S}^\bullet$-morphisms from S; its arrows are the commutative triangles in $\mathcal{S}^\bullet$. Consider the endofunctor P_S on $S/\mathcal{S}^\bullet$, assigning to each $f \in S/\mathcal{S}^\bullet$ the surjective part of $(\Sigma \times f)_! \circ \sigma$.



(21)

The familiar limit constructions [4, 19] yield a terminal P_S-coalgebra $t : \tau \to \mathsf{P}_S(\tau)$ in $S/\mathcal{S}^\bullet$. As any terminal coalgebra, t is an isomorphism. Since P_S is defined up to isomorphism, we can take $\mathsf{P}_S(\tau) = \tau$.



(22)

The codomain $\widehat{S}$ of τ thus carries a canonical $\mathcal{C}$-structure and τ itself appears as an epimorphism $S \twoheadrightarrow \widehat{S}$ in $\mathcal{C}$.

Now note that any epimorphism $\gamma : S \twoheadrightarrow S'$ in $\mathcal{C}$ actually carries a unique P_S-coalgebra structure $g : \gamma \to \mathsf{P}_S(\gamma)$ induced by the fact that $\mathsf{P}_S(\gamma)$ is the coimage of $\sigma \circ \gamma = (\Sigma \times \gamma)_! \circ \sigma$. Since τ is terminal among such coalgebras, there is a unique coalgebra homomorphism $\psi : \gamma \to \tau$. Of course, the underlying function $\psi : S' \to \widehat{S}$ preserves ι and satisfies $\psi \circ \gamma = \tau$. Since τ is an epi, ψ is. Moreover, using the fact that γ is epimorphic, one easily shows that $\psi : S' \twoheadrightarrow \widehat{S}$ is a $\mathcal{C}$-morphism. This shows that $\widehat{S}$ is a quotient, in a unique way, of any given quotient S' of S in $\mathcal{C}$. We have thus proved that

[4]This terminology should not be taken too seriously. Algebras $TX \to X$ for a monad T indeed do capture a great deal of algebra; but "algebras" $EX \to X$ for just any endofunctor E do not, and owe their name only to the similarity with the former. Of course, "coalgebras" $X \to EX$ are just dual to "algebras", and have little to do with, say, coalgebras on modules.

Proposition 4.8 *The terminal coalgebra for the endofunctor* P_S*, defined above, is the couniversal quotient of* S *in* $\mathcal{C}$*.*

Finally, if S is a reachable transition system, its couniversal quotient $\widehat{S}$ is obviously irredundant. In other words, by restricting the above construction to $\mathcal{A}^{rt}$, we get the irredundant representatives.

In fact, the inclusion $\mathcal{A}^{rt} \hookrightarrow \mathcal{A}^{t}$, as well as its coreflection, preserves the bisimulation morphisms. Proposition 4.7 thus restricts to an equivalence between $\mathcal{A}^{rt}_{\sim}$ and a coreflective subcategory $\mathcal{C}^r$ of $\mathcal{C}$. One could colocalize the above constructions in $\mathcal{C}^r$ from the beginning, but then the equalizers, used in the terminal coalgebra construction, will not be created by the forgetful functor to $\mathcal{S}^\bullet$ any more.

5 Reflecting processes in automata

We have described a construction $\widehat{(-)}$, providing an irredundant quotient $\tau_P : P \twoheadrightarrow \widehat{P}$ for an arbitrary reachable transition system P. The couniversality of τ as a bisimulation quotient means that its kernel is the largest bisimulation on P, i.e.

$$x' \sim x'' \quad \Longleftrightarrow \quad \tau_P(x') = \tau_P(x'') \tag{23}$$

It follows that a morphism $\varphi : P \to Q$ is sober if and only if $\tau_P(x') = \tau_P(x'')$ implies $\tau_Q \circ \varphi(x') = \tau_Q \circ \varphi(x'')$. Since τ_P is the coequalizer of its kernel, there is thus a unique morphism $\widehat{\varphi} : \widehat{P} \to \widehat{Q}$ such that $\widehat{\varphi} \circ \tau_P = \tau_Q \circ \varphi$.

$$\begin{array}{ccccc} K & \overset{x'}{\underset{x''}{\rightrightarrows}} & P & \overset{\tau}{\twoheadrightarrow} & \widehat{P} \\ & & {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\widehat{\varphi}} \\ & & Q & \underset{\tau}{\twoheadrightarrow} & \widehat{Q} \end{array} \tag{24}$$

In this way, the construction $\widehat{(-)}$ becomes a functor from the category $\dot{\mathcal{A}}^{rt}$ of reachable transition systems and sober morphisms to the category $\mathcal{I}$ of irredundant automata. In fact, the latter is obviously included in the former, and $\widehat{(-)} : \dot{\mathcal{A}}^{rt} \longrightarrow \mathcal{I}$ is left adjoint to the inclusion: the couniversal quotients τ form the unit of the adjunction. Precomposing $\widehat{(-)}$ with the reflection of $\dot{\mathcal{A}}^{r} \to \dot{\mathcal{A}}^{rt}$ renders $\mathcal{I}$ as a reflective subcategory of $\dot{\mathcal{A}}^{r}$.

Furthermore, the construction $\widehat{(-)}$ can be extended into a functor from the poset-enriched category $\ddot{\mathcal{A}}$ of all automata and sober simulations — from which $\mathcal{P}$ was actually obtained as a universal quotient. For an arbitrary automaton P, the general situation is

$$\begin{array}{ccccc} P & \overset{\varepsilon}{\hookleftarrow} & P^r & & \\ {\scriptstyle\eta}\downarrow & & \downarrow{\scriptstyle\eta} & \searrow & \\ & & & & \widehat{P^{rt}} \\ & & & \nearrow{\scriptstyle\tau} & \\ P^t & \underset{\varepsilon}{\hookleftarrow} & P^{rt} & & \end{array} \tag{25}$$

where P^r its reachable part and P^t the induced transition system, while η and ε denote the adjunction data of (co)reflections displayed on (3). All the arrows displayed on (25) are bisimilation morphisms, and $\widehat{P^{rt}}$ is thus bisimilar to all of the shown automata.

The functor $\widetilde{(-)} : \ddot{\mathcal{A}} \longrightarrow \mathcal{I}$ should now take each automaton P to $\widetilde{P} = \widehat{P^{rt}}$. A sober simulation

$P \overset{p}{\leftarrow} R \overset{q}{\rightarrow} Q$ will then be transformed into the morphism $\widetilde{\langle p, q\rangle} : \widetilde{P} \to \widetilde{Q}$ according to the following scheme.

$$
\begin{array}{ccccccc}
P & \xleftarrow{\varepsilon} & P^r & \xrightarrow{\eta} & P^{rt} & \xrightarrow{\tau} & \widetilde{P} \\
{\scriptstyle p}\uparrow & & {\scriptstyle p^r}\uparrow & & \uparrow{\scriptstyle p^{rt}} & & {\scriptstyle \widehat{p^{rt}}}\uparrow\cong \\
R & \xleftarrow{\varepsilon} & R^r & \xrightarrow{\eta} & R^{rt} & \xrightarrow{\tau} & \widetilde{R} \\
{\scriptstyle q}\downarrow & & {\scriptstyle q^r}\downarrow & & \downarrow{\scriptstyle q^{rt}} & & {\scriptstyle \widehat{q^{rt}}}\downarrow \\
Q & \xleftarrow[\varepsilon]{} & Q^r & \xrightarrow[\eta]{} & Q^{rt} & \xrightarrow[\tau]{} & \widetilde{Q}
\end{array}
\qquad \widetilde{\langle p,q\rangle} : \widetilde{P} \to \widetilde{Q} \tag{26}
$$

According to theorem 3.3(a), relation $\langle p, q\rangle$ is a sober simulation if and only if p is a bisimulation morphism and q is sober. Clearly, the reflections and coreflections involved here preserve this property. Therefore, the arrow $\widehat{q^{rt}}$ is well-defined, while $\widehat{p^{rt}}$ must be an isomorphism, according to lemma 4.2. So we define $\widetilde{\langle p, q\rangle}$ as the composite $\widehat{q^{rt}} \circ \widehat{p^{rt}}^{-1}$.

Postcomposed with the equivalence $\mathcal{I} \xrightarrow{\sim} \mathcal{P}$, the functor $\widetilde{(-)} : \ddot{\mathcal{A}} \longrightarrow \mathcal{I}$ brings us back to the quotient $\ddot{\mathcal{A}} \longrightarrow \mathcal{P}$ from subsection 2.3. By lemma 2.3, the restriction of this functor to process simulations is a weak equivalence. The same must hold for the analogous restriction of $\widetilde{(-)} : \ddot{\mathcal{A}} \longrightarrow \mathcal{I}$. But here there is a canonical section $\mathcal{I} \hookrightarrow \ddot{\mathcal{A}}$. The restriction of $\widetilde{(-)}$ to process simulations is therefore a *strong* equivalence.

On the other hand, the restriction of $\widetilde{(-)}$ to reachable automata turns out to be left adjoint to the inclusion $\mathcal{I} \hookrightarrow \ddot{\mathcal{A}}^r$. This may seem strange, since the hom-sets of $\ddot{\mathcal{A}}^r$ are nontrivially enriched, whereas $\mathcal{I}$ is an ordinary category. But lemma 4.2 implies that any sober simulation from a reachable P to $\widetilde{Q}$ must be saturated. By lemma 2.4, the hom-set $\ddot{\mathcal{A}}^r(P, \widetilde{Q})$ must be discrete. Looking at (26) one easily establishes the bijective correspondence of the sober simulations $P \leftarrow R \to \widetilde{Q}$ and the morphisms $\widetilde{P} \to \widetilde{Q}$.

Theorem 5.1 *The category of irredundant automata $\mathcal{I}$ is strongly equivalent with the category of all automata with process simulations. Moreover, it is a reflective subcategory of reachable automata with sober simulations, or just with sober morphisms.*

6 Future work

The proposed picture of processes as irredundant automata is meant to provide grounds for a structural analysis of process calculi, in terms of universal, eventually logical properties. Some steps towards a logical interpretation of the basic operations on processes have been made in [16, 17], following the trail of Abramsky's interpretation of process calculus as a calculus of relations [1]. The crucial parts of the structure of important interaction categories has been derived from regular logic, by quotienting modulo bisimilarity. The adjunctions described in section 5 seem to play an important, although not yet fully understood role.

On the other hand, the developed representation of processes modulo strong bisimilarity can serve as a basis for representing some richer notions of a process. Processes modulo the weak and the branching bisimilarity are treated in [18].

A Appendix: Subterminal objects

Definition A.1 *An object Z is* subterminal *if there is at most one arrow $X \to Z$ from any object X of the same category.*

On the other hand, if there is *at least* one arrow $X \to Z$ from every X, then Z is *weakly terminal.* An object is thus terminal if and only if it is both subterminal and weakly terminal.

In the presence of a terminal object, the subterminal objects are just its subobjects. In the presence of binary products, a subterminal object Z can be recognized by the fact that the two projections $Z \times Z \to Z$ are equal, or that one of them is isomorphism, or that the diagonal $Z \to Z \times Z$ is isomorphism. The products are the meets in the preorder of subterminal objects.

Proposition A.2 *In the presence of pullbacks and coequalizers, an object is subterminal if and only if it has no nontrivial quotients. If the coequalizers are stable under the pullbacks, it follows that $X \twoheadrightarrow Z$ is a terminal (i.e. couniversal) quotient of X if and only if Z is subterminal.*

B Appendix: The Beck-Chevalley over quasi-pullbacks

Consider a commutative square in a topos and its combined covariant-contravariant lifting to powersets:

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle t}\downarrow & & \downarrow{\scriptstyle s} \\ C & \xrightarrow[g]{} & D \end{array} \qquad\qquad \begin{array}{ccc} \wp A & \xleftarrow{f^*} & \wp B \\ {\scriptstyle t_!}\downarrow & & \downarrow{\scriptstyle s_!} \\ \wp C & \xleftarrow[g^*]{} & \wp C \end{array} \tag{27}$$

— where f^* and g^* are the inverse images (i.e., by the contravariant powerset functor), while $t_!$ and $s_!$ are the direct images (by the covariant powerset functor). The Beck-Chevalley condition says that the lifted square must commute whenever the original square is a pullback. We shall show that this commutativity actually characterizes the quasi-pullbacks. The logical explanation of this fact, and the idea for the proof below, can be found in the introductory remarks of [14]. It is not limited to toposes, but remains valid for any regular fibration satisfying function comprehension [15].

Proposition B.1 *A commutative square (s, f, g, t) in a topos is a quasi-pullback if and only if $t_! \circ f^* = g^* \circ s_!$.*

Proof. Let $C \stackrel{\tilde{t}}{\leftarrow} \tilde{A} \stackrel{\tilde{f}}{\rightarrow} B$ be the span obtained by pulling back s and g; and let $e : A \to \tilde{A}$ be the unique arrow such that $f = \tilde{f} \circ e$ and $t = \tilde{t} \circ e$.

If e is an epi, then $e_! \circ e^* = \mathrm{id}$. Using the Beck-Chevalley condition, we get

$$t_! \circ f^* = \tilde{t}_! \circ e_! \circ e^* \circ \tilde{f}^* = \tilde{t}_! \circ \tilde{f}^* = g^* \circ s_!.$$

Towards the converse, first note that the relation $\langle f, t\rangle_!(\top_A) \in \wp(B \times C)$ encodes the function $t_! \circ f^* \circ \{\} : B \to \wp C$, where $\{\} : B \to \wp B$ is the singleton map; whereas $(s \times g)^* \delta_!(\top_D) \in \wp(B \times C)$ similarly corresponds to $g^* \circ s_! \circ \{\} : B \to \wp C$. Therefore, $t_! \circ f^* = g^* \circ s_!$ holds if and only if $\langle f, t\rangle_!(\top_A) = (s \times g)^* \circ \delta_!(\top_D)$.

Using the Beck-Chevalley condition over the following pullbacks

$$\begin{array}{ccccc} A & \xrightarrow{e} & \tilde{A} & \xrightarrow{q} & D \\ {\scriptstyle \mathrm{id}}\downarrow & & \downarrow{\scriptstyle \langle \tilde{f}, \tilde{t}\rangle} & & \downarrow{\scriptstyle \delta} \\ A & \xrightarrow[\langle f, t\rangle]{} & B \times C & \xrightarrow[s \times g]{} & D \times D \end{array} \tag{28}$$

we calculate:

$$\langle f, t\rangle_!(\top_A) = (s \times g)^* \circ \delta_!(\top_D) = \langle \tilde{f}, \tilde{t}\rangle_! \circ q^*(\top_D) = \langle \tilde{f}, \tilde{t}\rangle_!(\top_{\tilde{A}}),$$

and hence

$$e_!(\top_A) = \langle \tilde{f}, \tilde{t}\rangle^* \circ \langle f, t\rangle_!(\top_A) = \langle \tilde{f}, \tilde{t}\rangle^* \circ \langle \tilde{f}, \tilde{t}\rangle_!(\top_{\tilde{A}}) = \top_{\tilde{A}}.$$

But this means that e is an epimorphism. $\square$

Acknowledgements. I am grateful to S. Abramsky, T. Maibaum and D. Murphy for friendly advice and references. Many thanks to Paul Johnson for careful reading and for detecting a basic error in an earlier version.

References

[1] S. Abramsky, Interaction categories, *manuscript* (1994); *extended abstract in:* G. Burn et al (eds.), Theory and Formal Methods 1993, *Workshops in Computing* (Springer 1993) 57–69

[2] P. Aczel, *Non-Well-Founded Sets*, Lecture Notes 14 (CSLI 1988)

[3] P. Aczel and P.F. Mendler, A final coalgebra theorem, in: *Category Theory and Computer Science*, D.H. Pitt et al. (eds.), Lecture Notes in Computer Science 389 (Springer 1989) 357–365

[4] M. Barr, Terminal coalgebras in well-founded set theory, *Theoret. Comput. Sci.* 114(1993) 299–315

[5] D.B. Benson and O. Ben-Shachar, Bisimulation of automata, *Information and Computation* 79(1988) 60–83

[6] J. van Bentham, Correspondence theory, in: *Handbook of Philosophical Logic*, D.M. Gabbay, ed., (Reidel 1984) 167–247

[7] J.A. Bergstra and J.W. Klop, Algebra of communicating processes with abstraction, *Theoret. Comput. Sci.* 37(1985) 77–121

[8] I. Castellani, Bisimulations and abstraction morphisms, *in:* H. Ehrig et al., eds., *Mathematical Foundations of Software Development, Vol.1: CAAP '85* Lecture Notes in Computer Science 185 (Springer 1986) 224–238

[9] P.J. Freyd and A. Scedrov, *Categories, Allegories*, North-Holland Mathematical Library 39 (North-Holland, 1990)

[10] R.J. van Glabbeek, *Comparative Concurrency Semantics and Refinement of Actions*, thesis (CWI, 1990)

[11] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation* (Addison-Wesley 1979)

[12] A. Joyal, M. Nielsen and G. Winskel, Bisimulation and open maps, *Proceedings of the Eight Symposium on Logic in Computer Science* (IEEE 1993) 418–427

[13] D. Park, *Concurrency and Automata on Infinite Sequences*, Lecture Notes in Computer Science 104 (Springer 1980)

[14] D. Pavlovic, Categorical interpolation: descent and the Beck-Chevalley condition without direct images, *in:* A. Carboni et al. (eds.), *Category Theory*, Lecture Notes in Mathematics 1488 (Springer, 1991) 306–326

[15] D. Pavlovic, Maps II: Chasing diagrams in categorical proof theory, *submitted*

[16] D. Pavlovic, Categorical logic of concurrency and interaction I: Synchronous processes, *in:* C. Hankin et al. (eds.), *Proceedings of the Second TFM Workshop, Cambridge 1994* (World Scientific 1995)

[17] D. Pavlovic, Categorical logic of concurrency and interaction II: Asynchronous processes, *in preparation*

[18] D. Pavlovic, Convenient Categories of Processes and Simulations II: Asynchronous Cases, *submitted*

[19] J.J.M.M Rutten and D. Turi, Initial algebra and final coalgebra semantics for concurrency, *in:* J.W. de Bakker et al. (eds.), *Proc. of the REX School*, Lecture Notes in Computer Science (Springer 1994)

[20] K. Segeberg, Decidability of S4.1, *Theoria* 34(1968) 7–20

[21] G. Winskel and M. Nielsen, Models for concurrency, *in:* S. Abramsky et al. (eds.), *Handbook of Logic in Computer Science*, vol. IV (Clarendon Press 1994) *to appear*