Specifying Interaction Categories
(extended abstract)

S. Abramsky* and D. Pavlović†
Department of Computing, Imperial College, London SW7 2BZ

Abstract

We analyse two complementary methods for obtaining categorical models of process calculi. They allow adding new features respectively to the captured notion of process and to the notion of type. By alternating these two methods, all the familiar examples, as well as some new interaction categories, can be derived from basic monoidal categories.

Using the proposed constructions, interaction categories can be built and analysed without fixing a set of axioms for them. They are thus approached *via* specifications, just like algebras are approached *via* equations and relations, independantly of the intrinsic characterisation of varieties.

1 Introduction

Interaction Categories [2] are proposed as a general, yet practical tool for reasoning about functional and concurrent computation. They are not meant to be a definitive formal system, but rather a task specification, suggesting a particular framework for a solution. The paradigm of processes as relations extended in time is taken as the conceptual basis for integrating type theory with process calculus, on the background of categorical structures. The interaction of processes is captured by composition.

We make a step towards determining the structure of Interaction Categories by analysing the ways in which they come about. It turns out that all existing examples, and several new ones, are obtained by alternating sequences of two *specification methods*, determining respectively the notion of process and the notion of type.

The notion of specification is here understood as a method of building a structure from given material. For instance, universal algebra is the method of

*Current address: Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ

†Current address: COGS, University of Sussex, Brighton BN1 9QH. This work was partially supported under ONR grant N00014-92-J-1974.

specifying by operations and equations; forcing is a method of specifying new models of set theory over the old ones. Note that the Birhkoff theorem, axiomatising categories that arise in universal algebra, as well as the Giraud theorem, providing the axioms for those those which arise from forcing, came only after extensive development of the corresponding specification methods. Thorough studies of the practice of specifying usually precede abstract characterisation of a class of structures.

2 Specifications and categories derived from them

The two specification methods that we are about to describe both begin from an arbitrary, possibly degenerate interaction category $\mathcal{R}$. The first of them yields a category with the same objects as $\mathcal{R}$ but with morphisms capturing a richer notion of process, while the second one refines the type structure, but leaves the morphisms essentially unchanged.

2.1 Specifying processes

Definition 2.1 *A functor* $\mathsf{h} : \mathcal{R} \to \mathcal{Q}$ *between monoidal categories* [16, sec. 1.1.] *is said to be* lax monoidal *if it is given with a natural family*

$$\begin{array}{rcl} \mu_{AB} & : & \mathsf{h}A \otimes \mathsf{h}B \longrightarrow \mathsf{h}(A \otimes B) \ \textit{and an arrow} \\ \eta & : & \top \longrightarrow \mathsf{h}\top, \end{array}$$

which are coherent in the sense that for all A, B, C *the following diagrams commute*

$$\begin{array}{ccc} \mathsf{h}A \otimes \mathsf{h}B \otimes \mathsf{h}C & \xrightarrow{\mu\otimes\mathrm{id}} & \mathsf{h}(A \otimes B) \otimes \mathsf{h}C \\ \Big\downarrow{\scriptstyle \mathrm{id}\otimes\mu} & & \Big\downarrow{\scriptstyle \mu} \\ \mathsf{h}A \otimes \mathsf{h}(B \otimes C) & \xrightarrow[\mu]{} & \mathsf{h}(A \otimes B \otimes C) \end{array} \qquad \begin{array}{ccc} \mathsf{h}A & \xrightarrow{\eta\otimes\mathrm{id}} & \mathsf{h}\top \otimes \mathsf{h}A \\ \Big\downarrow{\scriptstyle \mathrm{id}\otimes\eta} & \searrow{\scriptstyle \mathrm{id}} & \Big\downarrow{\scriptstyle \mu} \\ \mathsf{h}A \otimes \mathsf{h}\top & \xrightarrow[\mu]{} & \mathsf{h}A \end{array} \tag{1}$$

— *with monidal isomorphisms* $(A\otimes B)\otimes C \cong A\otimes(B\otimes C)$ *and* $A\otimes\top \cong A \cong \top\otimes A$ *omitted for simplicity.* h *is said to be* (strong) monoidal *when* μ *and* η *are isomorphisms.*

Remark. Recall that a monoidal category $\mathcal{R}$ is $\star$-autonomous [5] if and only if it is self-dual, and the duality[1] $(-)^\star : \mathcal{R}^{op} \to \mathcal{R}$, $A \cong A^{\star\star}$, induces the

[1] As a small contribution to the notational confusion in this context, we denote the two monoidal structures of $\star$-autonomous categories by $(\otimes, \top)$ and $(⅋, \bot)$. The usual notations $(\times, 1)$ and $(+, 0)$ for products and coproducts are often replaced respectively with $(\&, 1)$ and $(\oplus, 0)$.

cotensor $B \multimap C$ in the form $(B \otimes C^\star)^\star$, thus making $\mathcal{R}$ autonomous (i.e. closed symmetric monoidal [16, sec. 1.5]). Now a $\star$-autonomous $\mathcal{R}$ satisfies the MIX rule [13] if and only if the duality functor $(-)^\star$ is lax monoidal; and it is compact closed [17] if and only if the duality is monoidal.

Process specifications. All these concepts readily generalise to enriched categories [16]. *Process specifications* are meant to be $\mathsf{Pos}_\bot$-enriched lax monoidal functors $\mathsf{h} : \mathcal{R} \to \mathsf{Pos}_\bot$, where $\mathcal{R}$ is at least autonomous and $\mathsf{Pos}_\bot$ is the category of posets with the bottom; morphisms are the bottom preserving monotone maps, and the monoidal structure is induced by the cartesian products. The base $\mathcal{R}$ is to be thought of as a category of abstract sets and relations. The functor h defines a notion of process by specifying for each set the processes typed by it. They are partially ordered by computational power, i.e. their ability to *simulate* one another.

For simplicity, we presently leave the enriched aspects aside, and formalize process specifications as functors to Set. Formally, though, any monoidal category would do.

Construction. The category of processes $\mathcal{R}_h$ induced by a specification $\mathsf{h} : \mathcal{R} \to \mathsf{Set}$ will have the same objects as $\mathcal{R}$. To define the morphisms, we compose the internal hom-functor $\multimap : \mathcal{R}^{op} \times \mathcal{R} \longrightarrow \mathcal{R}$ with $\mathsf{h} : \mathcal{R} \longrightarrow \mathsf{Set}$. Hence

$$\mathcal{R}_\mathsf{h}(A, B) \quad = \quad \mathsf{h}(A \multimap B). \tag{2}$$

The structure of $\mathcal{R}$ then readily lifts to $\mathcal{R}_\mathsf{h}$. The identity on A is obtained using the transposition $\ulcorner \mathrm{id}_A \urcorner : \top \longrightarrow (A \multimap A)$ of the corresponding identity in $\mathcal{R}$:

$$\mathrm{id}_A \quad = \quad \mathsf{h}\ulcorner \mathrm{id}_A \urcorner(\eta). \tag{3}$$

The composite of $f \in \mathcal{R}_\mathsf{h}(A, B)$ and $g \in \mathcal{R}_\mathsf{h}(B, C)$, i.e. of $\langle f, g\rangle \in \mathsf{h}(A \multimap B) \times \mathsf{h}(B \multimap C)$, becomes

$$f \,; g \quad = \quad \mathsf{h}m(\mu\langle f, g\rangle), \tag{4}$$

where $m : (A \multimap B) \otimes (B \multimap C) \longrightarrow (A \multimap C)$ is the internal composition in $\mathcal{R}$. The autonomous structure on the objects is inherited directly, while the arrow part is first internalised. For instance, the functor $X \otimes (-)$ on $\mathcal{R}$ induces a family of arrows from $A \multimap B$ to $(X \otimes A) \multimap (X \otimes B)$ in $\mathcal{R}$, the h-image of which is a family of functions from $\mathcal{R}_\mathsf{h}(A, B)$ to $\mathcal{R}_\mathsf{h}(X \otimes A, X \otimes B)$.

Note that $\mathcal{R}_\mathsf{h}$ comes with a functor $J = J_\mathsf{h} : \mathcal{R} \to \mathcal{R}_\mathsf{h}$. It is identity on the objects, and it maps $f \in \mathcal{R}(A, B)$ to $Jf = \mathsf{h}\ulcorner f \urcorner(\eta)$, where $\ulcorner f \urcorner \in \mathcal{R}(\top, A \multimap B)$ is the transpose of f. It is autonomous by the very definition of the autonomous structure in $\mathcal{R}_\mathsf{h}$.

The other way around, any functor $F : \mathcal{R} \to \mathcal{Q}$ induces a representation $\mathsf{h} = \mathsf{h}_F : \mathcal{R} \to \mathsf{Set}$, with $\mathsf{h}A = \mathcal{Q}(\top, FA)$. If F is monoidal, h is lax monoidal, with $\eta = \mathrm{id}_\top$ and μ_{AB} induced by tensoring the arrows $\top \to FA$ and $\top \to FB$ to get $\top \longrightarrow FA \otimes FB \cong F(A \otimes B)$. If F is autonomous, we can construct $\mathcal{R}_\mathsf{h}$ as above, and it will be isomorphic with $\mathcal{Q}$ if and only if F is bijective on objects. In fact, any essentially surjective F induces a weak equivalence $F' : \mathcal{R}_{\mathsf{h}} \to \mathcal{Q}$, with $F = (J\,; F')$. We spell out just the 1-dimensional part of the underlying 2-adjunction. Note that it extends to $\mathcal{V}$-enriched categories for any monoidal $\mathcal{V}$ in place of Set.

Fix an autonomous $\mathcal{R}$ and consider the category $\mathcal{R}/\mathsf{Bij}$ of bijective on objects, autonomous functors out of it. A morphism from such an $F : \mathcal{R} \to \mathcal{Q}$ to $G : \mathcal{R} \to \mathcal{P}$ will be an autonomous functor $M : \mathcal{Q} \to \mathcal{P}$, satisfying $(F\,; M) = G$ (and necesssarily bijective on objects too).

On the other hand, let $[\mathcal{R}, \mathsf{Set}]_{lax\otimes}$ be the category of lax monoidal functors and lax monoidal transformations. A natural transformation $\varphi : \mathsf{h} \to \mathsf{h}'$ is said to be lax monoidal if $\eta\,; \varphi_\top = \eta'$ and $(\mu_{AB}\,; \varphi_{A\otimes B}) = \big((\varphi_A \times \varphi_B)\,; \mu'_{AB}\big)$.

Proposition 2.2 $\mathcal{R}/\mathsf{Bij} \quad \simeq \quad [\mathcal{R}, \mathsf{Set}]_{lax\otimes}$

2.2 Specifying types

Definition 2.3 *Let $\mathcal{R}$ be a category and $\mathcal{B}$ a bicategory* [7]*. A* lax functor *$\mathsf{P} : \mathcal{C} \to \mathcal{B}$ is an assignment for each object A of $\mathcal{R}$ of an object $\mathsf{P}A$ in $\mathcal{B}$ and for each arrow $f : A \to B$ of a 1-cell $\mathsf{P}f : \mathsf{P}A \to \mathsf{P}B$ in $\mathcal{B}$. Furthermore, P comes equipped with the 2-cells*

$$\begin{aligned} \mu_{fg} &: \mathsf{P}f\,; \mathsf{P}g \longrightarrow \mathsf{P}(f\,; g) && \textit{for every composable } f \textit{ and } g, \textit{ and} \\ \eta_A &: id_{\mathsf{P}A} \longrightarrow \mathsf{P}(\mathrm{id}_A) && \textit{for every object } A, \end{aligned}$$

satisfying coherence conditions similar to (1).

The lax monoidal functors from 2.1 are just lax functors between monoidal categories, regarded as bicategories with one object.

Type specifications. To refine the type structure of an interaction category $\mathcal{R}$, we assign for each type $A \in \mathcal{R}$ a set $\mathsf{P}A$ of new "properties", or "predicates" over it. Putting them all together, we construct a new interaction category $\mathcal{R}^{\mathsf{P}}$. No new processes are added: an $\mathcal{R}^{\mathsf{P}}$-morphism from $\alpha \in \mathsf{P}A$ to $\beta \in \mathsf{P}B$ will be just an $\mathcal{R}$-morphism $f : A \to B$, mapping the elements that satisfy α to those that satisfy β. Which αs and βs does f connect in this way will be specified by a relation $\mathsf{P}f \subseteq \mathsf{P}A \times \mathsf{P}B$. Clearly, such relations will usually not satisfy more than

$$\alpha\{\mathsf{P}f\}\beta \wedge \beta\{\mathsf{P}g\}\gamma \implies \alpha\{\mathsf{P}(f\,; g)\}\gamma \tag{5}$$

$$\alpha = \alpha' \implies \alpha\{\mathsf{P}(\mathrm{id})\}\alpha' \tag{6}$$

where $\alpha\{\mathsf{P}f\}\beta$ abbreviates $\langle\alpha, \beta\rangle \in \mathsf{P}f$. A type specification thus turns out to be a lax functor P from an interaction category $\mathcal{R}$ to the $\mathsf{Pos}_\perp$-category Rel of sets and relations.

Extracting from such a specification $\mathsf{P} : \mathcal{R} \to \mathsf{Rel}$ an interaction category $\mathcal{R}^{\mathsf{P}}$ is not essentially more complicated than extracting $\mathcal{R}_{\mathsf{h}}$ in 2.1, but it has very general background and deep conceptual roots.

Comprehension for categories. Consider the bicategory Span: its objects are sets, and a morphism from A to B is a pair of functions $A \leftarrow M \to B$. A 2-cell to another such pair $A \leftarrow M' \to B$ is just a function $\varphi : M \to M'$, commuting with the pairs. Given a span $B \leftarrow N \to C$, the composite $A \leftarrow (M\,;N) \to C$ is obtained by calculating a pullback of $M \to B$ and $B \leftarrow N$. Identities will clearly be in the form $A \stackrel{\mathrm{id}}{\leftarrow} A \stackrel{\mathrm{id}}{\to} A$. A span $A \stackrel{a}{\leftarrow} M \stackrel{b}{\to} B$ can also be viewed as an $A \times B$-matrix of sets, with $\langle a, b\rangle^{-1}(i,j)$ as the (i,j)-th entry. The 2-cells are obviously just entry-wise families of functions. The described composition then corresponds the usual matrix multiplication, using the set-theoretical sums and products.

Now any lax functor $\mathsf{P} : \mathcal{R} \to \mathsf{Span}$ induces the *total category* $\int_{\mathcal{R}} \mathsf{P}$, defined:

$$|\int_{\mathcal{R}} \mathsf{P}| \quad = \quad \sum_{X \in |\mathcal{R}|} \mathsf{P}X \tag{7}$$

$$\int_{\mathcal{R}} \mathsf{P}\left(\langle A, \alpha\rangle, \langle B, \beta\rangle\right) \quad = \quad \sum_{f \in \mathcal{R}(A,B)} \alpha\{\mathsf{P}f\}\beta \tag{8}$$

where $\alpha\{\mathsf{P}f\}\beta$ is the (α,β)-th entry of the matrix $\mathsf{P}f$. The composite of $\langle f, \varphi\rangle : \langle A, \alpha\rangle \longrightarrow \langle B, \beta\rangle$ and $\langle g, \psi\rangle : \langle B, \beta\rangle \longrightarrow \langle C, \gamma\rangle$ in $\int_{\mathcal{R}} \mathsf{P}$, is $\Big\langle (f\,;g),\, \mu_{fg}^{\alpha\gamma}(\beta, \varphi, \psi)\Big\rangle$, where

$$\mu_{fg}^{\alpha\gamma} \quad : \quad \sum_{\beta \in \mathsf{P}B} \alpha\{\mathsf{P}f\}\beta \times \beta\{\mathsf{P}g\}\gamma \longrightarrow \alpha\left\{\mathsf{P}(f\,;g)\right\}\gamma \tag{9}$$

is the (α,γ)-th component of the 2-cell μ_{fg}. The identity on $\langle A, \alpha\rangle$ is $\langle \mathrm{id}_A, \eta_A(\alpha)\rangle$.

While the total category comes with the obvious projection $I : \int_{\mathcal{R}} \mathsf{P} \longrightarrow \mathcal{R}$, any functor $F : \mathcal{Q} \longrightarrow \mathcal{R}$ (say, between small categories) induces a lax functor $\mathsf{P}_F : \mathcal{R} \to \mathsf{Span}$, with an isomorphism $F' : \mathcal{Q} \to \int_{\mathcal{R}} \mathsf{P}_F$ satisfying $F = (F'\,;I)$. The lax functor P_F sends each $A \in \mathcal{R}$ to the set $\mathsf{P}A = \{\alpha \in \mathcal{Q} | F\alpha = A\}$, and each arrow $f : A \to B$ to the $\mathsf{P}A \times \mathsf{P}B$-matrix of sets

$$\alpha\{\mathsf{P}f\}\beta \quad = \quad \{\varphi \in \mathcal{Q}(\alpha,\beta) | F\varphi = f\}. \tag{10}$$

The described correspondence extends to the equivalence

Proposition 2.4[2] $\mathsf{Cat}/\mathcal{R} \quad \simeq \quad [\mathcal{R}, \mathsf{Span}]_{lax}$

[2] Although this correspondence definitely seems too basic to be unknown, we remain unable to find any reference to it in the literature or folklore. A considerably more complicated one, relating $\mathsf{Cat}/\mathcal{R}$ with the normalised lax functors from $\mathcal{R}$ to categories and profunctors, is often mentioned, though, and has been known for long [8].

between the category of functors to $\mathcal{R}$, with commutative triangles as morphisms, and the category of lax functors $\mathcal{R} \to \mathsf{Span}$ and the functional lax transformations. A lax transformation $\varphi : \mathsf{P} \to \mathsf{Q} : \mathcal{R} \to \mathsf{Span}$ is a family of matrices $\varphi_A : \mathsf{P}A \nrightarrow \mathsf{Q}A$ with a coherent 2-cell $(\mathsf{P}f\,;\varphi_B) \longrightarrow (\varphi_A\,;\mathsf{Q}f)$ for every $f : A \to B$. It is said to be *functional* if all components φ_A are functions.

The establisned equivalence extends in various directions. By dropping the functionality requirement, and varying the notion of lax transformation on the right-hand side, one gets various interesting classes of morphisms on the left-hand side: indexed profunctors and anafunctors [18], and a categorical form of simulations. On the other hand, it restricts to the Conduché correspondence [23], to the Grothendieck construction [15], and so on, until it boils down to the familiar correspondence $\mathsf{Set}/R \simeq [R, \mathsf{Set}]$ of the functions to a set R and the R-indexed sets — and, finally, to the *comprehension scheme* $\mathsf{Sub}/R \cong [R, \Omega]$, connecting the subobjects of R with the predicates over it. Indeed, just as the extension $\{x \in R | p(x)\} \hookrightarrow R$ can be obtained as a pullback of the truth $t : 1 \to \Omega$ along the predicate $p : R \to \Omega$, the construct $\int_{\mathcal{R}} \mathsf{P} \longrightarrow \mathcal{R}$ can be obtained as a pullback along $\mathsf{P} : \mathcal{R} \to \mathsf{Span}$ of the obvious projection $\mathsf{t} : \mathsf{Span}^\bullet \longrightarrow \mathsf{Span}$, where $\mathsf{Span}^\bullet$ is the total category of the identity on Span.

To restrict to the lax functors $\mathsf{P} : \mathcal{R} \to \mathsf{Rel}$, note that a relation $R \hookrightarrow A \times B$ is a jointly monic span $A \leftarrow R \to B$, i.e. a matrix of 0s and 1s. The canonical functor $\mathsf{Span} \to \mathsf{Rel}$ is thus obtained by taking monic images of spans, or, in terms of matrices of sets, by reducing each nonempty entry to 1. The category $[\mathcal{R}, \mathsf{Rel}]_{lax}$ is thus a reflective subcategory of $[\mathcal{R}, \mathsf{Span}]_{lax}$. On the left-hand side of 2.4 the category $\mathsf{Fait}/\mathcal{R}$ of *faithful* functors to $\mathcal{R}$ corresponds to it: by (8), $\int_{\mathcal{R}} \mathsf{P} \longrightarrow \mathcal{R}$ will be faithful if and only if each $\alpha\{\mathsf{P}f\}\beta$ is just 0 or 1.

The interaction category specified by $\mathsf{P} : \mathcal{R} \to \mathsf{Rel}$ will be $\mathcal{R}^{\mathsf{P}} = \int_{\mathcal{R}} \mathsf{P}$.

2.3 Lifting the structure

In principle, the signature of an Interaction Category combines linear logic with the delay monads, in an enriched setting. In the present paper, we can only comment on the very first aspect.

The linear structure of $\mathcal{R}$ lifts to $\mathcal{R}_{\mathsf{h}}$ in a fairly straightforward way. First of all, $\mathcal{R}_{\mathsf{h}}$ is $\star$-autonomous (resp. compact closed) if and only if $\mathcal{R}$ is. Namely, any endofunctor D on $\mathcal{R}$ lifts to an endofunctor on D_{h} on $\mathcal{R}_{\mathsf{h}}$: the arrow part is again the h-image of the obvious family $(A \multimap B) \longrightarrow (DA \multimap DB)$. In this way, the duality lifts from $\mathcal{R}$ to $\mathcal{R}_{\mathsf{h}}$.

Moreover, any natural transformation between lifted endofunctors lifts too — along the functor $\mathcal{R} \to \mathcal{R}_{\mathsf{h}}$. Monads and comonads thus induce monads and comonads. Recall that a *bang* is a monoidal comonad $! : \mathcal{R} \to \mathcal{R}$ the coalgebras of which are $\otimes$-comonoids. This can be expressed by natural transformations $e_A : !A \to \top$ and $d_A : !A \to !A \otimes !A$, imposintg the required structure [9].

A bang thus lifts from $\mathcal{R}$ to $\mathcal{R}_{\mathsf{h}}$. However, the couniversal bang, sending each object to the corresponding cofree $\otimes$-comonoid, may loose its property in lifting.

Finally, using just definition (2), one easily shows that the (weak) products *and* coproducts are preserved and thus created by the functor $\mathcal{R} \to \mathcal{R}_{\mathsf{h}}$ as soon as the specification $\mathsf{h} : \mathcal{R} \to \mathsf{Set}$ preserves the (weak) products. However, we shall see that usually does not. Process specifications alone thus yield categories with few limits and colimits. Adding more types corrects this.

Lifting structures along type specifications is less straightforward, although quite uniform. Looking at the correspondence from proposition 2.4, one sees that any, say, binary functorial operation $\diamond$, preserved by $\int_{\mathcal{R}} \mathsf{P} \longrightarrow \mathcal{R}$, corresponds to a functional lax transformation $\mathsf{P}A \times \mathsf{P}B \stackrel{\diamond}{\longrightarrow} \mathsf{P}(A \diamond B)$, with $\langle A, \alpha\rangle \diamond \langle B, \beta\rangle = \langle A \diamond B, \alpha \diamond \beta\rangle$. In order to lift $\diamond$ from $\mathcal{R}$ to $\mathcal{R}^{\mathsf{P}}$, we must thus specify the corresponding transformations. This is where we depart from the degeneracies of $\mathcal{R}$.

3 Examples

The idea is to start from a simple model $\mathcal{R}$, and successively refine it by specifying

$$\mathcal{R} \longrightarrow \mathcal{R}_{\mathsf{h}_1} \longleftarrow (\mathcal{R}_{\mathsf{h}_1})^{\mathsf{P}_1} \longrightarrow ((\mathcal{R}_{\mathsf{h}_1})^{\mathsf{P}_1})_{\mathsf{h}_2} \longleftarrow (((\mathcal{R}_{\mathsf{h}_1})^{\mathsf{P}_1})_{\mathsf{h}_2})^{\mathsf{P}_2} \longrightarrow \cdots$$

The view of processes as relations in time suggests that any category of relations could be taken as the base $\mathcal{R}$. Namely, the calculus of relations as jointly monic spans can be developed not just over sets but over more general categories $\mathcal{C}$ [12]. The obtained category $\mathsf{Rel}(\mathcal{C})$ is always compact closed, but varying $\mathcal{C}$ allows additional structure on *actions*.

3.1 Synchrony

The simplest case is of course $\mathsf{Rel} = \mathsf{Rel}(\mathsf{Set})$. Let the process specification $\mathsf{s} : \mathsf{Rel} \to \mathsf{Set}$ assign to every set A the poset $\mathsf{s}A$ of nonempty, prefix-closed sets of finite strings from A. These strings are to be thought of as "the elements of A extended in time", so that the elements of $\mathsf{s}A$ become "the subsets of A extended in time". Algebraically, they can be presented as one-sided multiplicative systems of the free monoid A^*, i.e., the complements of the one-sided ideals of A^*.

The arrow part of s will map a relation $A \leftarrow R \to B$ to the function $\mathsf{s}R : \mathsf{s}A \to \mathsf{s}B$, defined

$$\mathsf{s}R(S) \quad = \quad \{t \in B^* | \exists s \in S. sR^* t\}, \tag{11}$$

where $A^* \leftarrow R^* \rightarrow B^*$ is the componentwise extension of R to strings. The lax monoidal structure consists of the function $\mu_{AB} : \mathsf{s}A \times \mathsf{s}B \longrightarrow \mathsf{s}(A \otimes B)$, where

$$\mu_{AB}(S,T) \quad = \quad \{u \in (A \otimes B)^* | \pi_A^*(u) \in S \wedge \pi_B^*(u) \in T\}, \tag{12}$$

and $\eta \in \mathsf{s}1$ consisting of all finite strings of $\bullet \in 1$.

The category $\mathsf{sproc} = \mathsf{Rel_s}$, obtained by the construction from 2.1, is a rudimentary interaction category of synchronous processes, modulo the trace equivalence. Finer notions of behaviour are obtained by taking as the elements of $\mathsf{s}A$ transition systems, or A-labelled trees, rather than just the traces $S \subseteq A^*$. Definitions (11) and (12) readily extend. Working modulo bisimilarity complicates matters [19, 20], but everything goes through.

The synchronous interaction category SProc [2] is obtained by a further type specification $\mathsf{S} : \mathsf{sproc} \rightarrow \mathsf{Rel}$. Its object part will actually be the same as for the above process specification. Its arrow part should take the process $U \in \mathsf{sproc}(A, B)$ to the relation $\mathsf{S}A \leftarrow \mathsf{S}U \rightarrow \mathsf{S}B$ defined

$$S\{\mathsf{S}U\}T \quad \Longleftrightarrow \quad \forall u \in U.\ \pi_A^*(u) \in S \wedge \pi_B^*(u) \in T \tag{13}$$

If sproc is taken modulo bisimilarity, the process U in this definition should be replaced by the corresponding set of traces.

The category SProc is thus $(\mathsf{Rel_s})^{\mathsf{S}}$. This order of specifying can be changed, as one can easily see by constructing the pullback of the functors $\mathsf{Rel} \longleftarrow \mathsf{sproc} \longrightarrow \mathsf{SProc}$, obtained from specifications. Although it is intuitively simpler to first specify the notion of process, the advantage of first specfying the types is that the biproducts and the cofree comonoids of SProc — neither of which are present in sproc — can then be traced back to Rel.

3.2 Asynchrony

To capture the asynchrony, one can start from the calculus of relations $\mathsf{Rel}^\bullet$ developed over the category $\mathsf{Set}^\bullet$ of *pointed* sets, all containing a fixed element $\bullet$, which all functions must preserve. $\bullet$ represents the *idle* action, which allows processes to wait. $\mathsf{Set}^\bullet$ is the Kleisli category for the monad $1+(-) : \mathsf{Set} \rightarrow \mathsf{Set}$, but it is sometimes useful to view it as the category of sets and *partial* functions. $\mathsf{Rel}^\bullet = \mathsf{Rel}(\mathsf{Set}^\bullet)$ can thus be presented either as the full subcategory of Rel spanned by the objects in the form $1 + A$, or as the category of sets and partial relations. A partial relation $A \leftharpoondown R \rightharpoonup B$ actually boils down to a triple $\langle R_A, R_\times, R_B\rangle$, where $R_\times \hookrightarrow A \times B$ is an ordinary binary relation, while $R_A \hookrightarrow A$ and $R_B \hookrightarrow B$ the parts where R is undefined. The tensor and the cotensor are $A \otimes B = A + B + A \times B$, and the embedding $1+(-) : \mathsf{Rel}^\bullet \rightarrow \mathsf{Rel}$ preserves them. The weak biproducts $A + B$ are also preserved, and note that $\mathsf{Rel}^\bullet$ does not have the strong ones, which is reflected in the asynchronous interaction categories.

To specify $\mathsf{as}^\bullet : \mathsf{Rel}^\bullet \rightarrow \mathsf{Set}$, identify $\mathsf{Rel}^\bullet$ with its image in Rel and note that

$\bullet$ must be the unit of any monoid in $\mathsf{Set}^\bullet$. Rather than $(1+A)^*$, the free monoid over $1+A$ is thus $1+A^+$, where A^+ consists of all *nonempty* strings from A.

The object part of $\mathsf{as}^\bullet$ thus takes $1+A$ to the set of prefix-closed subsets of $1+A^+$, each containing $\bullet$. The arrow part is defined using the monoid homomorphism $\widetilde{(-)} : (1+A)^* \longrightarrow 1+A^+$, which removes $\bullet$ from all nontrivial strings, and induces the weak equivalence $s \approx t \iff \widetilde{s} = \widetilde{t}$. A relation $1+A \leftarrow R \rightarrow 1+B$ now goes to the function $\mathsf{as}^\bullet R : \mathsf{as}^\bullet A \longrightarrow \mathsf{as}^\bullet B$, defined

$$\mathsf{as}^\bullet R(S) \quad = \quad \{t \in 1+B^+ | \exists s \in S.\ s \approx R^* \approx t\}. \tag{14}$$

In words, a string t belongs to $\mathsf{as}^\bullet R(S)$ if there is a string s in S such that s and t can be filled up with sequences of $\bullet$ in such a way that they become componentwise R-related.

By a similar trick, the function $\mu_{AB} : \mathsf{as}^\bullet A \times \mathsf{as}^\bullet B \longrightarrow \mathsf{as}^\bullet(A \otimes B)$ shuffles the strings:

$$\mu_{AB}(S,T) \quad = \quad \left\{u \in ((1+A)\times(1+B))^+ \,|\widetilde{\pi_A^*}(u) \in S \wedge \widetilde{\pi_B^*}(u) \in T\right\} \tag{15}$$

An element of $\mu_{AB}(S,T)$ is obtained by taking some $s \in S$ and $t \in T$, possibly of different length, interpolating $\bullet$ in them at will, to get $s' = \alpha_1 \ldots \alpha_n$ and $t' = \beta_1 \ldots \beta_n$, and then forming $u = \langle \alpha_1, \beta_1 \rangle \ldots \langle \alpha_n, \beta_n \rangle$. The unit is $\eta = \{\bullet\}$.

The asynchronous interaction category $\mathsf{as}^\bullet\mathsf{proc} = \mathsf{Rel}^\bullet_{\mathsf{as}^\bullet}$ is obtained as before. A version depicting a finer notion of behaviou can again obtained using $(1+A)$-labelled trees or transition systems, this time modulo weak or branching bisimilarity. A full fledged asynchronous category $\mathsf{AS}^\bullet\mathsf{Proc}$, with *weak* biproducts and a *weakly* couniversal bang, is obtained by adding more types along a specification $\mathsf{AS}^\bullet : \mathsf{as}^\bullet\mathsf{proc} \longrightarrow \mathsf{Rel}$, similar to S from section 3.1, but relaxed modulo $\approx$.

The original asynchronous category ASProc [2, sec. 5] is obtained in the same way, but using relations in place of partial functions, i.e. starting from $\mathsf{Req} = \mathsf{Rel}(\mathsf{Rel})$ rather than $\mathsf{Rel}^\bullet = \mathsf{Rel}(\mathsf{Set}^\bullet)$. Req is the category of sets and the *partial equivalence* relations on $A+B$ as the morphisms from A to B. Namely, a relation $A \leftarrow\!\!\!\!+ \ R \!+\!\!\!\!\rightarrow B$ in Rel boils down to a jointly surjective pair $A \rightarrow R \leftarrow B$ in $\mathsf{Set}^\bullet$. Alternatively, Req can be viewed as the full subcategory of Rel spanned by the power sets $\wp A$. The tensor preservation along the embedding $\wp : \mathsf{Req} \rightarrow \mathsf{Rel}$ boils down to the exponential laws $\wp(A+B) \cong \wp A \times \wp B$ and $\wp 0 = 1$.

The specification $\mathsf{as} : \mathsf{Req} \rightarrow \mathsf{Set}$ assigns to each A the set of nonempty prefix closed sets of sequences from $\wp^+ A = \wp A - \emptyset$. The empty set is deleted because it plays the role of $\bullet$. The resulting category $\mathsf{asproc} = \mathsf{Req}_{\mathsf{as}}$ compares to $\mathsf{as}^\bullet\mathsf{proc}$ just as Req compares to $\mathsf{Rel}^\bullet$. For instance, bang comonads are precluded by the fact that any functor $! : \mathsf{Req} \rightarrow \mathsf{Req}$ with a natural family $e_A : !A \rightarrow 0$ must be trivial.

The structure of actions can be further enriched using other monads on Set. E.g., consider the one sending A to $1+A+A$. (If its unit is chosen to include

A in $1+A+A$ as the first copy, then the multiplication should send the first two As from $1+(1+A+A)+(1+A+A)$ to $1+A+A$ in order, and twist the last two of them.) Besides the idling $\bullet$, this monad captures the input/output distinction — between the elements of the two copies of A. The Kleisli category $\mathsf{Set}^{\overline{\bullet}}$ for this monad can now be viewed as the category of sets, with pairs $\langle f, F\rangle$ as morphisms from A to B, where f is a partial function $A \rightharpoonup B$ and F is a subset of A. The composite of $\langle f, F\rangle : A \to B$ and $\langle g, G\rangle : B \to C$ consists of the usual composite of partial function $(f\,;g)$, accompanied with the set $\big(F \cap \varphi^{-1}(G)\big) \cup \big(\overline{F} \cap \varphi^{-1}(\overline{G})\big)$, where $\overline{F}, \overline{G}$ denote the complements. The free monoid $A^{\overline{*}}$ over A in $\mathsf{Set}^{\overline{\bullet}}$ will be the quotient of $1+(A+A)^+$ satisfying $\alpha\overline{\alpha} = \bullet$ for all $\alpha \in A$, with $\overline{(-)} : A+A \longrightarrow A+A$ denoting the twist map. All monoids in $\mathsf{Set}^{\overline{\bullet}}$ are thus groups — which means that any computation can be "consumed" and "internalized" as $\bullet$. One is thus led to consider the *infix closed* sets $S \subseteq A^{\overline{*}}$, i.e. such that

$$\alpha s \overline{\alpha} \in S \quad \Longrightarrow \quad s \in S \tag{16}$$

for any $\alpha \in A$. They correspond to normal subgroups of $A^{\overline{*}}$ roughly like the prefix closed sets correspond to the ideals of A^*, the underlying idea being that, in reversible time, computations develop in two directions.

The specification $\mathsf{as}^{\overline{\bullet}} : \mathsf{Rel}^{\overline{\bullet}} \to \mathsf{Set}$, where $\mathsf{Rel}^{\overline{\bullet}} = \mathsf{Rel}(\mathsf{Set}^{\overline{\bullet}})$, will now assign to each A the set of the infix closed subsets of $A^{\overline{*}}$. The arrow part can be formally defined just as for $\mathsf{as}^{\bullet}$ — but the kernel of $\widetilde{(-)} : (1+A+A)^* \longrightarrow A^{\overline{*}}$ will be much larger and instead of sequences of $\bullet$, we shall be interpolating more general strings of the input and output actions, that reduce to $\bullet$.

For finer notions of behaviour, instead of infix closed sets, one could use transition systems without a distinguished initial state, or labelled acyclic graphs, modulo the corresponding notion of bisimilarity...

3.3 Coherence

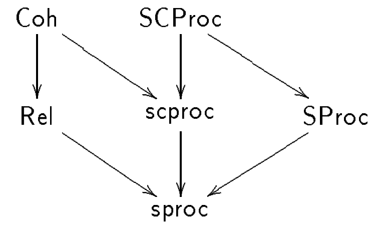
Each of categories constructed so far can be refined by first extending Rel, say, by the notion of *coherence*. It can be introduced by lax functor $\mathsf{C} : \mathsf{Rel} \to \mathsf{Rel}$, assigning to each set A the set of all symmetric, irreflexive binary relations on it. A relation $A \leftarrow R \rightarrow B$ now induces $\mathsf{C}A \leftarrow \mathsf{C}R \rightarrow \mathsf{C}B$, defined

$$\Phi\{\mathsf{C}R\}\Psi \quad \Longleftrightarrow \quad \forall \alpha\alpha' \in A\beta\beta' \in B.\ (\alpha R\beta \wedge \alpha' R\beta') \Rightarrow (\alpha\Phi\alpha' \Rightarrow \beta\Psi\beta') \tag{17}$$

The total category $\mathsf{Rel}^{\mathsf{C}}$ will be the familiar category Coh of coherence spaces [14]. By imposing on each set of traces $S \subseteq A^*$ (or on labelled trees, or transition systems) the coherence requirement

$$s\alpha, s\alpha' \in S \quad \Longrightarrow \quad \alpha\Phi\alpha' \tag{18}$$

for all $\alpha \neq \alpha' \in A$, all previously described specifications lift to Coh, and yield interaction categories with a grain of true concurrency. It is interesting to notice that already the synchronous ones can be specified in many different, meaningful ways.



(19)

3.4 Games

Categories of games are specifed starting from *signed* sets $\mathsf{Set}_\pm$. A signed set A is a pair $\langle A_-, A_+\rangle$, of ordinary sets; a signed function $f : A \to B$ is an ordinary partial function $f : A_+ + B_- \rightharpoonup B_+ + A_-$. To compose it with $g : B \to C$, i.e. $g : B_+ + C_- \rightharpoonup C_+ + B_-$, follow the images of each $x \in A_+$ along the tower $A_+ \overset{f}{\rightharpoonup} B_+ \overset{g}{\rightharpoonup} B_- \overset{f}{\rightharpoonup} B_+ \overset{g}{\rightharpoonup} \cdots$. If an image ever leaves B and lands in A_- or in C_+, it will be the value of $(f\,;g)$ at x. Otherwise, $(f\,;g)$ remains undefined at x. To define $(f\,;g)$ on C_-, follow the tower $C_- \overset{g}{\rightharpoonup} B_- \overset{f}{\rightharpoonup} B_+ \overset{g}{\rightharpoonup} B_- \cdots$. The identity on $A = \langle A_-, A_+\rangle$ is obviously the identity on $A_+ + A_-$. The obtained category is compact closed, with the structure

$$A \otimes B = \langle A_- + B_-, A_+ + B_+\rangle, \tag{20}$$
$$A^\star = \langle A_+, A_-\rangle. \tag{21}$$

In fact, the free compact closed categories are effectively described in terms of $\mathsf{Set}_\pm$ [17, sec. 3] — in fact, by a type specification over it. A further, somewhat more complicated refinement yields the free $\star$-autonomous categories.

A basic category of games is obtained by a type specification $\mathsf{G} : \mathsf{Set}_\pm \to \mathsf{Rel}$, sending each A to the set of all nonempty prefix closed subsets of $(A_- \times A_+)^*$. The functions $\mathsf{G}A \times \mathsf{G}B \overset{\otimes}{\longrightarrow} \mathsf{G}(A \otimes B)$ and $\mathsf{G}A \times \mathsf{G}B \overset{\multimap}{\longrightarrow} \mathsf{G}(A^\star \otimes B)$ shuffle these sets, the latter in the subtle way described in [3]. A signed function $f : A \to B$ then induces $\mathsf{G}A \leftarrow \mathsf{G}f \rightarrow \mathsf{G}B$, relating Ψ and Φ if and only if f yields a *history free strategy* for $\Phi \multimap \Psi$ [3].

The history sensitive strategies can now be introduced in a process specification over the total category $\mathsf{Set}_\pm^{\mathsf{G}}$. Furthermore, the winning positions can be added in a further type specification, or just extending the specification G. Clearly, G can also be extended to include the equivalence relations on positions, essential for [4]. The relations $\mathsf{G}f$ will be supplied with the requirement that f preserves the equivalences on the games being related. The category from [4] will follow from an additional process specification, identifying the equivalent strategies.

References

[1] S. Abramsky, Specification structures and propositions-as-types for concurrency, *talk* at the CONFER meeting in Paris, April 1995

[2] S. Abramski et al., Interaction categories and the foundations of the typed concurrent programming, to appear in the proceedings of BANFF

[3] S. Abramsky and R. Jagadeesan, Games and full completeness for multiplicative linear logic, *J. Symbolic Logic*

[4] S. Abramsky et al., Full abstraction for PCF, *submitted*

[5] M. Barr, *$\star$-Autonomous Categories*, Lecture Notes in Mathematics 752 (Springer 1979)

[6] M. Barr, $\star$-Autonomous categories and linear logic, *Math. Structures Comput. Sci.* 1/2(1991), 159–178

[7] J. Bénabou, Introduction to bicategories, in: *Reports of the Midwest Category Seminar I*, Lecture Notes in Mathematics 47 (Springer, 1967) 1–77

[8] J. Bénabou, 2-dimensional limits and colimits of distributors, abstract of a talk given in Oberwolfach (1972)

[9] G.M. Bierman, What is a categorical model of intuitionistic linear logic?, in: *Proceedings of Conference on Typed Lambda Calculus and Applications*, M. Dezani-Ciancaglini and G. Plotkin, eds., Lecture Notes in Computer Science 902 (Springer 1995)

[10] V. Danos and L. Regnier, The structure of multiplicatives, *Archive form Math. Logic* 28(1989) 181–203

[11] T. Fox, Coalgebras and cartesian categories, *Comm. Algebra*, 4/7(1976) 665–667

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

[13] A. Fleury and C. Retoré, The MIX rule, Unpublished note, 1990

[14] J.-Y. Girard et al., *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science 7 (Cambridge Univ. Press 1989)

[15] A. Grothendieck, Catégories fibrées et descente, Exposé VI, *Revêtements Etales et Groupe Fondamental (SGA1)*, Lecture Notes in Mathematics 224 (Springer, 1971) 145–194

[16] G.M. Kelly, *Basic Concepts of Enriched Category Theory*, L.M.S. Lecture Notes 64 (Cambridge Univ. Press 1982)

[17] G.M. Kelly and M.L. Laplaza, Coherence for compact closed categories, *J. Pure Appl. Algebra* 19(1980) 193–213

[18] M. Makkai, Avoiding the axiom of choice in general category theory, to appear in *J. Pure Appl. Algebra*

[19] D. Pavlović, Categorical logic of concurrency and interaction I. Synchronous processes, in: *Theory and Formal Methods of Computing 1994*, C.L. Henkin et al., eds. (World Scientific 1995), 105–141

[20] D. Pavlović, Convenient categories of processes and simulations I: modulo strong bisimilarity, *Category Theory and Computer Science '95*, D.H. Pitt et al., eds., Lect. Notes in Comp. Science 953 (Springer, 1995), 3–24

[21] D. Pavlović, Maps I: relative to a factorisation system, *J. Pure Appl. Algebra* 99(1995), 9–34; Maps II: Chasing diagrams in categorical proof theory, *J. of the IGPL* 3/7(1995), 1–36

[22] R.A.G. Seely, Linear logic, $\star$-autonomous categories and cofree coalgebras, in: J. Gray and A. Scedrov (eds.), *Categories in Computer Science and Logic*, *Contemp. Math.* 92 (Amer. Math. Soc., 1989), 371–382

[23] R. Street, Conduché functors, *a hand written note*, dated 15 October 1986, 4 pp.