A categorical model of the Fusion calculus

Marino Miculan

Department of Mathematics and Computer Science, University of Udine
Via delle Scienze 206, I-33100 Udine, Italy. miculan@dimi.uniud.it

Abstract

We provide a *categorical presentation* of the Fusion calculus. Working in a suitable category of presheaves, we describe the syntax as initial algebra of a signature endofunctor, and the semantics as coalgebras of a "behaviour" endofunctor. To this end, we first give a new, congruence-free presentation of the Fusion calculus; then, the behaviour endofunctor is constructed by adding in a systematic way a notion of "state" to the intuitive endofunctor induced by the LTS. Coalgebras can be given a concrete presentation as "stateful indexed labelled transition systems"; the bisimilarity over these systems is a congruence, and corresponds to hyperequivalence. Then, we model the labelled transition system of Fusion by as *abstract categorical rules*. As a consequence, we get a semantics for the Fusion calculus which is both compositional and fully abstract: two processes have the same semantics iff they are bisimilar, that is, hyperequivalent.

1 Introduction

In recent years, Parrow and Victor's Fusion calculus [20] has emerged as a good foundational model for distributed computation paradigms like web-services and service oriented architectures (SOAs). Fusion calculus is often regarded as a variant of the π-calculus, but actually it has quite distinctive features; the most dramatic difference is that synchronisation does not yield a name substitution, but rather a *fusion* of names. Fused names have the same meaning and can be used interchangeably. Several typical concepts of web services and SOAs can be represented via fusions, such as negotiation between client requests and server capabilities, forwarders for objects migrating among locations, etc. Many recent specific calculi for SOAs are directly inspired by the Fusion calculus, such as CC-pi and JoCaML [4,9].

In this paper, we provide a *fully abstract categorical model* of the (finite) Fusion calculus. More precisely, we describe the syntax of processes as the initial algebra of a suitable signature endofunctor, and the semantics as coalgebras of a suitable "behaviour" endofunctor. These coalgebras can be concretely presented as a new kind of labelled transition systems, whose labels correspond to those of the Fusion calculus, but states are processes together with equivalences over names. Bisimilarity over these systems corresponds precisely to hyperequivalence. In fact, following the bialgebraic approach of Plotkin and Turi [26], we obtain a semantics which is both compositional and fully abstract, that is, two processes have the same semantics iff they are bisimilar, and hence, hyperequivalent.

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

In order to achieve these results, we have to solve several technical problems.

The first problem is that in the usual presentation of the Fusion calculus, processes are taken up-to a *structural congruence* [20], containing the usual laws such as *scope extrusion*. This is problematic because this syntax cannot be defined as a free algebra of some functor, which is a requirement for the construction of the fully-abstract, compositional semantics [26]. We circumvent this obstacle by giving in Section 2 a new presentation of the Fusion calculus without structural congruence, and where processes are taken only up-to α-conversion of bound names.

The presence of α-equivalence is not a problem, because languages with binders can be described as free algebras in suitable categories of set-valued functors, that is *presheaves*. In particular, the category $Set^{\mathbb{F}}$ of presheaves over $\mathbb{F}$, the category of finite sets and functions, is commonly used for dealing with languages defined by signatures with binding constructors "*a la* λ-calculus", and whose terms are taken up-to α-equivalence [7,14]. This is precisely the situation of the Fusion calculus, hence $Set^{\mathbb{F}}$ is the right category where this syntax can be abstractly presented as the presheaf $Proc$ which is the carrier of the initial algebra of a suitable endofunctor Σ_F (Section 4.1). We recall the main properties and constructions of $Set^{\mathbb{F}}$ in Section 3.

Now, in order to define a (fully abstract) semantics for the Fusion calculus, we need to define an object of *behaviours* in the same category where the syntax lies ($Set^{\mathbb{F}}$, in our case). This object must be defined as the (carrier of the) final coalgebra νB of a suitable endofunctor B; in this way, the semantics is the unique morphism from $Proc$ to νB given by terminality of νB (and initiality of $Proc$) [26].

The "behaviour endofunctor" B is typically of the form $BX = \wp_f(Act \times X)$, where Act is an object of *actions*, corresponding to the labels of the operational semantics of the calculus; then, B-coalgebras correspond to (finitely branching) labelled transition systems with labels in Act [2]. This construction can be carried out also in category of functors, by taking $\wp_f$ the pointwise finite powerset, as in [6].

However, there are two problems in applying this approach to the Fusion calculus. First, the behaviour functor B using the pointwise finite power-object $\wp_f$ on $Set^{\mathbb{F}}$ does not preserve weak pullbacks, which is required the construction of the "universal" semantics. We solve the problem by using the $\tilde{K}$*-finite power object* construction [10,16], which preserves weak pullbacks (Section 3.3).

The second issue is more subtle, and strictly related to the design choices behind the Fusion calculus. Using the construction above, we can indeed define an endofunctor D on $Set^{\mathbb{F}}$ such that processes of the Fusion calculus are D-coalgebras (see Section 4.2), but it turns out that the bisimulation over these coalgebras do not correspond to the intended equivalence of the Fusion calculus, that is, hyperequivalence. The problem is that the Fusion calculus "contains actions akin to updating a shared state" [20]: states are *equivalences* between names, which can be updated (i.e. extended) by fusions. Notably, this "shared state" *is not* explicitly represented in the labelled transition system of Fusion calculus. In this way, inference rules are simpler but, on the other hand, an *ad hoc* definition of bisimulation is needed.

In the categorical setting, we cannot tinker with the notion of bisimulation; hence we ought to accommodate the notion of behaviour, that is, the behaviour endofunctor. In Section 4.2 we will tackle this problem by adding the notion of state to the endofunctor D in a systematic way, that is, by lifting it along the adjunction

underlying the "state monad" $GX = (X \times E)^E$, for E a suitable object of states. Thus, the resulting "stateful" behavior functor is $B(X) = (D(X \times E))^E$. This approach is general and can be applied with any notions of global state. In the case of the Fusion calculus, the presheaf of states at n is the set of all possible equivalences of n names, that is, the set of coequalizers over n (Section 3.2). Moreover, in Section 4.3 we will present a concrete presentations of coalgebras of this endofunctor as *stateful indexed labelled transition system*, i.e., transition systems whose nodes are triples (n, P, e) where n is a set of names, $P \in P_n$ is a process and $e \in E_n$ is a state. Also coalgebraic bisimulations over the behaviour functor B can be given a concrete definition as a indexed family of relations. Using this characterization, in Section 4.4 we will show that coalgebraic bisimilarity coincides with hyperequivalence.

Finally, in Section 5 we complete our development by providing the "mathematical operational semantics" for the Fusion calculus. We translate the new labelled transition system introduced in Section 2 into an "abstract categorical rule", that is, a natural transformation between two endofunctors on $Set^{\mathbb{F}}$, using the syntactic and behaviour functors defined in Section 4. This presentation yields a fully-abstract (bialgebraic) semantics, that is, two processes have the same semantics iff they are B_F-bisimilar and hence hyperequivalent.

Final remarks and directions for future work are in Section 6.

2 The Fusion Calculus

In this section we give the syntax, semantics and bisimulations of the Fusion calculus [20]. For our aims, a *monadic* version of the calculus without replication will suffice; the results we present can be generalized routinely to the polyadic version.

The processes are defined by the following grammar: [1]

$$P, Q ::= \mathbf{0} \mid zx.P \mid \bar{z}x.P \mid P|Q \mid (x)P$$

where x is bound in $(x)P$. Processes are taken up to α-equivalence.

The semantics of Fusion is defined by a labelled transition relation of the form $P \xrightarrow{\alpha} Q$, whose labels are

$$\alpha ::= xy \mid \bar{x}y \mid x(z) \mid \bar{x}(z) \mid \mathbf{1} \mid x{=}y$$

The *input* xy, *output* $\bar{x}y$, the *empty fusion* $\mathbf{1}$ and the *fusion* $x{=}y$ are called *free actions*; the others are called *bound actions*, because z is bound. [2]

In Figure 1 we report (a monadic variant of) the labelled transition system given in [20]. In the conclusion of the rule *Open*, x is bound in the label and in the target P' (like in the "open" rule of the π-calculus). We use u as a wildcard for z and $\bar{z}$.

This system uses a structural congruence $\equiv$ to identify syntactically different presentations of the same agent. At the moment, the theory of mathematical operational semantics cannot be applied to semantics using structural congruences; therefore, in Figure 2 we present a new, congruence-free semantics of Fusion. The difference with the previous one is that, in place of the congruence rule, there is

[1] Sum and fusion prefix can be easily encoded in this fragment.

[2] In the polyadic version, objects of communications are lists of names and fusions are lists of equivalences.

$$Pref\ \frac{-}{\alpha.P \xrightarrow{\alpha} P} \qquad Com\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'} \qquad Par\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

$$Open\ \frac{P \xrightarrow{ux} P',\ u \notin \{x, \bar{x}\}}{(x)P \xrightarrow{u(x)} P'} \qquad Pass\ \frac{P \xrightarrow{\alpha} P',\ x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)P'} \qquad Scope\ \frac{P \xrightarrow{\{x=y\}} P'}{(x)P \xrightarrow{\mathbf{1}} P'\{y/x\}}$$

$$Congr\ \frac{P \equiv P'\ P' \xrightarrow{\alpha} Q'\ Q' \equiv Q}{P \xrightarrow{\alpha} Q} \qquad (P|Q)|R \equiv P|(Q|R) \qquad P|Q \equiv Q|P \qquad P|\mathbf{0} \equiv P$$

$$(x)\mathbf{0} \equiv \mathbf{0} \qquad (x)(y)P \equiv (y)(x)P \qquad P|(x)P \equiv (x)(P|Q) \text{ if } x \notin fn(P)$$

Fig. 1. Labelled transition system with structural congruence for the Fusion calculus.

$$Pref\ \frac{-}{\alpha.P \xrightarrow{\alpha} P}$$

$$Com\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\{x=y\}} P'|Q'} \qquad Par_l\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad Par_r\ \frac{P \xrightarrow{\alpha} P'}{Q|P \xrightarrow{\alpha} Q|P'}$$

$$Open\ \frac{P \xrightarrow{ux} P',\ u \notin \{x, \bar{x}\}}{(x)P \xrightarrow{u(x)} P'} \qquad Pass\ \frac{P \xrightarrow{\alpha} P',\ x \notin n(\alpha)}{(x)P \xrightarrow{\alpha} (x)P'} \qquad Scope\ \frac{P \xrightarrow{\{x=y\}} P'}{(x)P \xrightarrow{\mathbf{1}} P'\{y/x\}}$$

$$Close_l\ \frac{P \xrightarrow{u(x)} P',\ Q \xrightarrow{\bar{u}y} Q'}{P|Q \xrightarrow{\mathbf{1}} P'\{y/x\}|Q'} \qquad Close_r\ \frac{P \xrightarrow{ux} P',\ Q \xrightarrow{\bar{u}(y)} Q'}{P|Q \xrightarrow{\mathbf{1}} P'|Q'\{x/y\}}$$

$$Close\ \frac{P \xrightarrow{u(x)} P',\ Q \xrightarrow{\bar{u}(x)} Q'}{P|Q \xrightarrow{\mathbf{1}} (x)(P'|Q')}$$

Fig. 2. Labelled transition system without structural congruence for the Fusion calculus.

a symmetric rule for $|$, and three new rules $Close_l$, $Close_r$, $Close$, allowing for π-calculus-like input, bound output and "scope extrusions".

This semantics corresponds to that in Figure 1, in the sense that they derive the same transitions, up-to structural congruence, as state by the following result.

Theorem 2.1 (i) *If $P \xrightarrow{\alpha} Q$ in the system of Figure 1, then there exist P', Q' such that $P' \equiv P$, $Q' \equiv Q$, and $P' \xrightarrow{\alpha} Q'$ in the system of Figure 2.*

(ii) *If $P \xrightarrow{\alpha} Q$ in the system of Figure 2, then $P \xrightarrow{\alpha} Q$ in the system of Figure 1.*

Proof. (i) By induction on the derivation of $P \xrightarrow{\alpha} Q$ with the system in Figure 1. The only interesting case is when this derivation ends with an application of the $(Congr)$ rule. Let $P \equiv P_1$, $Q \equiv Q_1$ and $P_1 \xrightarrow{\alpha} Q_1$ with the system in Figure 1. By inductive hypothesis, there exist $P_1' \equiv P_1$ and $Q_1' \equiv Q_1$ such that $P_1' \xrightarrow{\alpha} Q_1'$ with the system in Figure 2. Then, by transitivity of $\equiv$, it is $P \equiv P_1'$ and $Q \equiv Q_1'$.

(ii) (Par_r) is clearly derivable from (Par) and $(Congr)$. It suffices to prove that $(Close)$, $(Close_l)$, $(Close_r)$, are admissible in the system of Figure 1. We will examine the case of $(Close)$, the others being similar (and simpler).

Let $P|Q \xrightarrow{\mathbf{1}} (x)(P'|Q')$ be a derivation ending with the $(Close)$ rule; this means that there are two derivations of $P \xrightarrow{u(x)} P'$ and $Q \xrightarrow{\bar{u}(x)} Q'$, where by α-equivalence we can suppose that $x \notin fn(u)$. Now, bound transitions can be introduced only using the $(Open)$ rule (possibly followed by some applications of (Par_l), (Par_r), $(Pass)$); hence, without loss of generality, we can suppose that $P = (x)P_1$ and $Q = (x)Q_1$, for some P_1, Q_1, such that there are two derivations $P_1 \xrightarrow{ux} P'$ and $Q_1 \xrightarrow{\bar{u}x} Q'$. Now we prove that $P|Q \xrightarrow{\mathbf{1}} (x)(P'|Q')$ is derivable without using $(Close)$, but using the congruence rule.

Let y be a fresh name; then $Q_1\{y/x\} \xrightarrow{\bar{u}y} Q'\{y/x\}$, and, by applying (Com), we have $P_1|Q_1\{y/x\} \xrightarrow{x=y} P'|Q'\{y/x\}$. Using $(Scope)$, we get $(y)(P_1|Q_1\{y/x\}) \xrightarrow{\mathbf{1}} (P'|Q'\{y/x\})\{x/y\}$. Since y is fresh, the source of this transition is equivalent to $P_1|(y)Q_1\{y/x\} = P_1|(x)Q_1$, and the target is exactly $P'\{x/y\}|Q'\{y/x\}\{x/y\} = P'|Q'$. By applying the $(Pass)$ rule we obtain $(x)(P_1|(x)Q_1) \xrightarrow{\mathbf{1}} (x)(P'|Q')$. Since $x \notin fn((x)Q_1)$, it is $(x)(P_1|(x)Q_1) \equiv (x)P_1|(x)Q_1 = P|Q$, and hence by an application of $(Congr)$ we are done. □

Finally, we recall the notions of *bisimilarity* and *hyperequivalence* from [20].

Definition 2.2 *We say that a name substitution σ (i.e., a function $\sigma : \mathcal{N} \to \mathcal{N}$)* agrees *with a fusion φ iff $\forall x, y : x\varphi y \iff \sigma(x) = \sigma(y)$.*

A fusion bisimulation *is a symmetric relation $\mathcal{S}$ between processes such that whenever $(P, Q) \in \mathcal{S}$, if $P \xrightarrow{\alpha} P'$ with $bn(\alpha) \cap fn(Q) = \emptyset$, then $Q \xrightarrow{\alpha} Q'$ and*

- *if α is a communication action: $(P', Q') \in \mathcal{S}$;*
- *if α is a fusion: $(P'\sigma, Q'\sigma) \in \mathcal{S}$, for some σ agreeing with α.*

P and Q are fusion bisimilar *if $(P, Q) \in \mathcal{S}$ for some fusion bisimulation $\mathcal{S}$.*

A hyperbisimulation *is a substitution-closed fusion bisimulation, i.e., an $\mathcal{S}$ such that $(P, Q) \in \mathcal{S}$ implies $(P\sigma, Q\sigma) \in \mathcal{S}$ for any substitution σ. P and Q are* hyperequivalent*, written $P \sim Q$, if they are related by a hyperbisimulation.*

Notice that only hyperequivalence is a congruence, while bisimilarity is not [20].

3 Categorical framework

3.1 Finite sets

We denote by $\mathbb{F}$ the (skeleton) category of finite sets and functions. We will use n, m to range over objects of $\mathbb{F}$, and σ, ρ over morphisms of $\mathbb{F}$. Objects of $\mathbb{F}$ represent sets of allocated names (or variables), and maps describe how name sets may evolve (e.g., new names can be added and existing names can coalesce, etc.); thus $\sigma : n \to m$ is a substitution of names in n with names in m. $\mathbb{F}$ has products and coproducts, defined as usual in *Set*. In particular, let us denote by $\delta : \mathbb{F} \to \mathbb{F}$ the functor $\delta n = n + 1$; the two injection maps are $old : n \to n + 1$ and $new : 1 \to n + 1$, which extend to natural transformations $old : Id_{\mathbb{F}} \to \delta$, $new : \mathcal{K}_1 \to \delta$ (where $\mathcal{K}_S$ is the constant presheaf $(\mathcal{K}_S)_n = S$). δ is a monad on $\mathbb{F}$: its unit is old, and the multiplication is $contr_n : n + 1 + 1 \to n + 1$, mapping the two added elements in the domain to the same element added in the codomain.

3.2 Presheaves

$Set^{\mathbb{F}}$ is the category of functors from $\mathbb{F}$ to Set, called *(covariant) presheaves over* $\mathbb{F}$; the application of this category to the representation of languages up-to α-conversion has been studied in depth in [14,7,21], among others. The basic idea is that a functor $A : \mathbb{F} \longrightarrow Set$ can be seen as a family of sets stratified according to different sets of names; e.g., A_n is the set of "elements of type A using names from n". For $\sigma : n \to m$ a substitution, the map $A_\sigma : A_n \to A_m$ describes how this substitution acts on the elements of A_n; for this reason, we will often denote $A_\sigma(a)$ as $a[\sigma]$.

As for any presheaf category, $Set^{\mathbb{F}}$ is a topos, hence it is cartesian closed. The structure of Set lifts to $Set^{\mathbb{F}}$, which has:

(i) *Products and coproducts*, computed pointwise (as with all limits and colimits in functor categories); e.g. $(P \times Q)_e = P_e \times Q_e$. The terminal object is the constant functor $\mathcal{K}_1 = \mathbf{y}(0)$, where $\mathbf{y} : \mathbb{F}^{op} \longrightarrow Set^{\mathbb{F}}$ is the (contravariant) Yoneda embedding: $\mathcal{K}_1(e) = 1$. The initial object is the constant functor $\mathcal{K}_\emptyset$.

(ii) A presheaf of *names* $N \triangleq \mathbf{y}(1)$. For n in $\mathbb{F}$, it is $N_n = \mathbb{F}(1, n) \cong n \in Set$; thus N is the embedding from $\mathbb{F}$ into Set.

(iii) A *dynamic allocation* functor $\delta : Set^{\mathbb{F}} \longrightarrow Set^{\mathbb{F}}$, induced by the δ on $\mathbb{F}$ as $\_ \circ \delta : Set^{\mathbb{F}} \longrightarrow Set^{\mathbb{F}}$.

(iv) *Exponentials* are defined as usual in functor categories:

$$(B^A)_n \triangleq Set^{\mathbb{F}}(\mathbf{y}(n) \times A, B)$$
$$(B^A)_\sigma(m) \triangleq \phi \circ (id_A \times (\_ \circ \sigma)) \quad \text{for } \sigma : n \to m \text{ in } \mathbb{F}, \phi : A \times \mathbb{F}(e, \_) \longrightarrow B$$

We will occasionally denote B^A also as $A \Rightarrow B$.

In particular, exponentials of representable functors have a nice definition:

Proposition 3.1 ([7,14]) *For all $n \in \mathbb{F}$, B in $Set^{\mathbb{F}}$: $B^{\mathbf{y}(n)} \cong B_{\_+n}$.*

Proof.
$$\begin{aligned} (B^{\mathbf{y}(n)})_m &= Set^{\mathbb{F}}(\mathbf{y}(m) \times \mathbf{y}(n), B) && \text{by definition of exponential} \\ &\cong Set^{\mathbb{F}}(\mathbf{y}(m+n), B) && \text{since } \mathbf{y} \text{ preserves coproducts} \\ &\cong B_{m+n} && \text{by Yoneda Lemma.} \end{aligned}$$
□

This allows us to point out a strict relation between N and δ:

Proposition 3.2 *$(\_)^N \cong \delta$, and hence $\_ \times N \dashv \delta$.*

Proof. Since $N = \mathbf{y}(1)$, by Proposition 3.1 we have $F^N \cong F_{1+\_} = F_{\delta(\_)} = \delta(F)$. The second part is an obvious consequence, because in CCC's it is always $\_ \times B \dashv (\_)^B$.□

The presheaf of equivalences For n in $\mathbb{F}$, E_n denotes the set of *equivalences over* n, that is, the possible partitions of n. Formally this is the set of coequalizers over n:

$$E_n \triangleq \{coeq(f, g) \mid k \in \mathbb{F}, f, g : k \rightrightarrows n\} \tag{1}$$

Thus the elements of E_n are surjective substitutions $e : n \twoheadrightarrow m$, up-to permutations of the codomain. Using the definition (1), an element $e \in E_n$ can be represented by a kernel pair $f, g : k \rightrightarrows n$, which, akin to fusions in the Fusion calculus, can be

explicitly described as a set of k name equations in n, $\{x_1{=}y_1, \ldots, x_k{=}y_k\}$, where $x_i = f(i)$ and $y_i = g(i)$. In the following we will use both the surjection and the fusion notations; in particular, the empty set denotes the identity substitution $\emptyset = id_n$.

The action of E on morphisms can be described as follows. For $\sigma : n \to m$, $e : n \twoheadrightarrow q$ in E_n: $E_\sigma(e) = coeq(\sigma f, \sigma g)$ where f, g are such that $e = coeq(f, g)$. In "fusion notation", it is $E_\sigma(\{x_1{=}y_1, \ldots, x_k{=}y_k\}) = \{\sigma(x_1){=}\sigma(y_1), \ldots, \sigma(x_k){=}\sigma(y_k)\}$.

We need to introduce some operations on equivalences.

- Given two equivalences $e_1 : n \twoheadrightarrow m_1, e_2 : n \twoheadrightarrow m_2$ in E_n, we define the *union* $e_1 \cup e_2 : n \twoheadrightarrow m$ in E_n as the pushout of e_1, e_2; concretely, this can be defined by union of the two sets of equations.
- For $e_1 = coeq(f_1, g_1) \in E_n, e_2 = coeq(f_2, g_2) \in E_m$, the *sum* $e_1 + e_2 \in E_{n+m}$ is $coeq(f_1{+}f_2, g_1{+}g_2)$; notice that no new equivalences are introduced. In particular, for $e \in E_n$ we denote by $e + 1 \in E_{n+1}$ the equivalence $e + \{*{=}*\}$; this extends to a natural transformation $\_ + 1 : E \to \delta E$.
- Finally, given $e : m \twoheadrightarrow k$ in E_m and $i : n \to m$, the *restriction* of e to n (along i), is defined as $e_{\restriction n} = ei : n \twoheadrightarrow ei(n)$ in E_n.

3.3 Finite power-object

Beside the rich structure shown above, for representing non-determinism of the Fusion calculus we need also a "well-behaved" power-object endofunctor. By "well-behaved" we mean that it can be used for defining functors for which final coalgebras exist, bisimulations are equivalence relations, and the bialgebraic construction of [26] can be carried out. Basically, this means that we require the power-object endofunctor to be finitary and to preserve weak pullbacks.

One may consider the "ω-bounded powerset endofunctor" $P_\omega : Set^{\mathbb{F}} \to Set^{\mathbb{F}}$, that is, the "pointwise finite powerset functor" [6]. Although this functor would suffice for expressing the non-determinism of the Fusion calculus, it is not satisfactory because it does not preserve weak pullbacks [15,24]. Another possibility is the *K-finite powerset*: in any topos $\mathcal{E}$, the endofunctor $K : \mathcal{E} \to \mathcal{E}$ is a subfunctor of the full power-object, such that $K(A)$ can be seen as "the object of (Kuratowski) finite subobjects of A". K is finitary and, if $\mathcal{E}$ is Boolean, it preserves weak pullbacks [15, Example 1.4]. Unfortunately, $Set^{\mathbb{F}}$ is not Boolean.

The problem with K-finite objects is that they are not closed under subobjects. To circumvent this problem, we can use another "finite powerobject" endofunctor $\tilde{K}$, introduced by P. Freyd in [10]; here we refer to [16, D5.4] for a detailed presentation.

Definition 3.3 [[16]] In a topos $\mathcal{E}$, for A an object of $\mathcal{E}$, let $\tilde{A}$ be its partial map classifier (that is, $\tilde{A} \rightarrowtail PA$ is the "object of subobjects of A with at most one element"). Then, $\tilde{K}(A)$ is the sub-join-semilattice of PA generated by $\tilde{A} \rightarrowtail PA$.

Proposition 3.4 *In any topos $\mathcal{E}$, the $\tilde{K}$ endofunctor preserves weak pullbacks.*

Proof. This can be proved following the same pattern in [15, Example 1.4] for the Kuratowski-finite power object; the proof applies to $\tilde{K}$ in $\mathcal{E}$ (even if $\mathcal{E}$ is not a Boolean topos) because $\tilde{K}(A)$ is always downward closed as a subobject of PA [16, Lemma D5.4.21]; hence every subobject of $\tilde{K}$-finite object is $\tilde{K}$-finite. □

Remark 3.5 It is worthwhile giving an explicit description of maps of the form $\phi : A \to \tilde{K}(B)$. Since $\tilde{K}(B)$ is generated by the subsingletons of B, the map ϕ corresponds to a finite family of partial maps $(\phi^i : A \rightharpoonup B)_{i \in I}$. Each partial map ϕ^i is a span $A \hookleftarrow D \xrightarrow{\phi^i} B$, whose left arm is mono [25]; D can be seen as the subobject of A where ϕ^i is defined. In our setting, i.e. $\mathcal{E} = Set^{\mathbb{F}}$, this is a span of natural transformations. By unfolding the naturality conditions, it is easy to check that this span of natural transformations corresponds to a family of maps $(\phi^i_n : A_n \to B_n + 1)_{n \in \mathbb{F}}$ subject to naturality "only where defined":

$$\text{for all } \sigma : n \to m, \text{ for all } a \in A_n, \text{ if } \phi^i_n(a) \in B_n \text{ then } \phi^i_m(A_\sigma(a)) = B_\sigma(\phi^i_n(a)) \quad (2)$$

Notice that such a family is not a natural transformation $A \to B + 1$, since it may happen that $\phi_n(a) = * \in 1$ and still $\phi_m(A_\sigma(a)) \in B_m$ (in general, partial maps $A \rightharpoonup B$ are not equivalent to total maps $A \to B + 1$; this holds in Boolean toposes, but $Set^{\mathbb{F}}$ is not Boolean). As a consequence, the map $\phi : A \to \tilde{K}(B)$, described by the family $(\phi^i : A \rightharpoonup B)_{i \in I}$, can be seen as a family of finite set-valued functions

$$(\phi_n : A_n \to \wp_f(B_n))_{n \in \mathbb{F}} \quad \text{by taking} \quad \phi_n(a) = \{\phi^i_n(a) \mid i \in I, \phi^i_n(a) \in B_n\}. \quad (3)$$

Again, this family is not a natural transformation $A \to \wp_f(B)$, but, by (2), the following "weak naturality" holds:

$$\text{for all } \sigma : n \to m, \text{ for all } a \in A_n : \{B_\sigma(b) \mid b \in \phi_n(a)\} \subseteq \phi_m(A_\sigma(a)). \quad (4)$$

3.4 Algebras and coalgebras in $Set^{\mathbb{F}}$

In order to use $Set^{\mathbb{F}}$ as an ambient category for giving algebraic and coalgebraic presentations of syntax and semantics of calculi with fusions, we need to use endofunctors on $Set^{\mathbb{F}}$ having initial algebras and final coalgebras.

Recall that *polynomial* endofunctors over a category $\mathbb{C}$ is the smallest class of endofunctors over $\mathbb{C}$ containing identity, all constant endofunctors, and closed under products and coproducts. It is well-known that any polynomial functor over Set is finitary, and hence has initial algebra and final coalgebra [2]. This result has been generalized to $Set^{\mathbb{F}}$ [7,14] in order to deal with syntax with binders up-to α-equivalence; to this end, the class of admitted endofunctor contains also δ. In our case, for defining the final coalgebras for the Fusion calculus, we will need also to consider the endofunctors $\tilde{K}, (\_)^E : Set^{\mathbb{F}} \to Set^{\mathbb{F}}$. Summarizing:

Proposition 3.6 (i) *Let $T : Set^{\mathbb{F}} \longrightarrow Set^{\mathbb{F}}$ be a polynomial functor possibly using also $\delta : Set^{\mathbb{F}} \to Set^{\mathbb{F}}$. Then T has an initial algebra.*

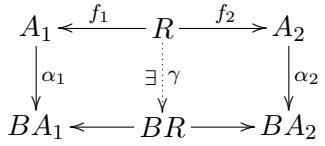
(ii) *Let $T : Set^{\mathbb{F}} \longrightarrow Set^{\mathbb{F}}$ be a polynomial functor possibly using also $\delta, \tilde{K}, (\_)^E : Set^{\mathbb{F}} \to Set^{\mathbb{F}}$. Then T has a final coalgebra.*

Proof. It is easy to check that δ and $\tilde{K}$ are finitary. We prove that $(\_)^E$ preserves limits. Let $F : \mathbb{D} \to Set^{\mathbb{F}}$ be a diagram; we prove that $(\lim(F))^E \cong \lim((\_)^E \circ F)$. In functor categories, limits and colimits are computed pointwise, so we have to prove that for $n \in \mathbb{F}$, $((\lim(F))^E)_n \cong \lim G_n$, where $G : \mathbb{F} \to Set^{\mathbb{D}}$ is defined by currying-uncurrying $(\_)^E \circ F : \mathbb{D} \to Set^{\mathbb{F}}$, that is, $(G_n)(d) \triangleq ((F(d))^E)_n$.

By expanding the definition of exponents in $Set^{\mathbb{F}}$, we have that $((\lim(F))^E)_n = Set^{\mathbb{F}}(\mathbf{y}(n) \times E, \lim(F)) \cong \lim Set^{\mathbb{F}}(\mathbf{y}(n) \times E, F)$, because the Hom functor preserves limits. By noticing that $Set^{\mathbb{F}}(\mathbf{y}(n) \times E, F) = (F(\_)^E)_n = G_n$, we are done. □

Finally, we recall the notion of *coalgebraic bisimulation*, as in [2].

Definition 3.7 For an endofunctor B on $Set^{\mathbb{F}}$, a *B-bisimulation* between two B-coalgebras (A_1, α_1) and (A_2, α_2) is a triple $(R, f_1 : R \to A_1, f_2 : R \to A_2)$ such that $\langle f_1, f_2 \rangle : R \to A_1 \times A_2$ is a strong monomorphism and there exists a B-coalgebra structure $\gamma : R \to BR$ such that f_1 and f_2 are B-coalgebra homomorphism, that is the diagram aside in $Set^{\mathbb{F}}$ commutes.



4 Modelling syntax and semantics of Fusion calculus

In this section we give a categorical representation of the syntax and semantics of the Fusion calculus, in the category $Set^{\mathbb{F}}$ defined in the previous section.

4.1 Algebraic syntax of the Fusion calculus

Let us first define the object of processes of the Fusion calculus. This is a free language with λ-like binders, up-to α-equivalence; thus we define the endofunctor Σ_F on $Set^{\mathbb{F}}$ as

$$\Sigma_F(A) \triangleq 1 + N \times N \times A + N \times N \times A + A \times A + \delta A \tag{5}$$

$$(\Sigma_F(A))_n = \overbrace{1}^{\mathbf{0}} + \overbrace{n \times n \times A_n}^{xy.a} + \overbrace{n \times n \times A_n}^{\bar{x}y.a} + \overbrace{A_n \times A_n}^{a|b} + \overbrace{A_{n+1}}^{(x)a}$$

Each summand correspond to a constructor of the language; in particular, agents with a bound name (i.e., of the form $(x)P$) and free names in n are represented as agents with free names in $n+1$, where the added name represent the bound name.

Then, the presheaf of processes of the Fusion calculus, denoted as $Proc$, is the (carrier of the) initial Σ_F-algebra, which exists in virtue of Proposition 3.6(i). Its algebra structure is denoted as $\alpha_F : \Sigma_F Proc \xrightarrow{\sim} Proc$. It is easy to see that:

Proposition 4.1 *For all sets of names m, $Proc_m = \{P \mid fn(P) \subseteq m\}$.*

We will denote by $T_F : Set^{\mathbb{F}} \longrightarrow Set^{\mathbb{F}}$ the monad of the free Σ_F-algebra, that is, $T_F(X) = \mu Y.X + \Sigma_F(Y)$. Clearly $Proc = T_F 0$.

4.2 Coalgebraic semantics of the Fusion calculus

The domain of the Fusion calculus is defined as the final coalgebra of a suitable "behaviour functor". As for other models of concurrency, this functor must account for finitely-branching non-determinism; to this end, we will use the $\tilde{K}$-finite power object defined in Section 3.3, which has the properties required for the successive developments. The behaviour functor must represent all possible actions (label) that a process can perform in its LTS; thus, looking at the semantics of Figure 2,

one could define this functor as

$$LX \triangleq \overbrace{N \times N \times X}^{\text{output}} + \overbrace{N \times N \times X}^{\text{input}} + \overbrace{N \times \delta X}^{\text{bound output}} + \overbrace{N \times \delta X}^{\text{bound input}} + \overbrace{X}^{\mathbf{1}} + \overbrace{N \times N \times X}^{\text{fusions}} \quad (6)$$

$$DX \triangleq \tilde{K}(LX) \quad (7)$$

Each component of functor L corresponds to a possible evolution of a presheaf X, viewed as a stratified set of processes. The various possible transitions are a label and a continuation; in particular,

- free transitions (input, output or fusion) have a pair of names as label, and a process in the same stage as continuation;
- empty fusions have no label, and a process in the same stage as continuation;
- bound transitions (bound input or output) have a single name as label, but the continuation process can use a fresh, local name (as denoted by the δ operator).

However, this semantics does not suffice for representing faithfully the Fusion calculus. The issue is that a fusion performed by a process can be not local, since it applies to any process running in parallel. As said in [20], a fusion "can be thought of as an update of a (not explicitly represented) shared state".

Hence, our model has to keep track of this global, shared state, which is an equivalence on the names currently defined, i.e., an element of E_n. A "stateful" behaviour of a process takes a given state $e \in E_n$, and produces a set of possible transitions, each of which yields a new state $e' \in E_{n'}$ alongside the continuation.

Thus we have to convert the "stateless" behaviour functor D (7) into a stateful one. We accomplish this by combining D with the "global state monad" [19]:

$$G : Set^{\mathbb{F}} \to Set^{\mathbb{F}}, \qquad GX = (X \times E)^E$$

D and G can be merged by decomposing the monad G in the corresponding adjunction $\_ \times E \dashv (\_)^E$, then, we define B_F as the lifting of D along this adjunction:

$$B_F X \triangleq (D(X \times E))^E \qquad B_F \circlearrowright Set^{\mathbb{F}} \underset{(\_)^E}{\overset{\_\times E}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} Set^{\mathbb{F}} \circlearrowright D \quad (8)$$

By definition of exponents in $Set^{\mathbb{F}}$, for n object of $\mathbb{F}$ we can write this explicitly as $(B_F X)_n = Set^{\mathbb{F}}(\mathbf{y}(n) \times E, D(X \times E))$. Thus, an element of $(B_F X)_n$ is a natural transformation $\phi : \mathbf{y}(n) \times E \to \tilde{K}L(X \times E)$, that is, a family of partial maps

$$(\phi^i : \mathbf{y}(n) \times E \rightharpoonup L(X \times E))_{i \in I}$$

According to Eq. 3, the behaviour of a process with n names, under the substitution $\sigma : n \to m$ and the equivalence e on m, is the union of all $\phi^i_m(\sigma, e)$ which are defined:

$$\phi_m(\sigma, e) = \{\phi^i_m(\sigma, e) \mid \phi^i_m(\sigma, e) \text{ defined}, i \in I\} \quad (9)$$

Each $\phi^i_m(\sigma, e)$, if defined, is a single stateful transition, i.e. a tuple containing the label, the continuation process (the element from X) and the new equivalence (over m names, if the transition is free, $m+1$ if it is bound). Naturality of ϕ^i ensures that

a transition which happens under the substitution $\sigma : n \to m$ is preserved under any further substitutions, but it is not required to correspond to some transition at n (i.e., $\phi^i_n(id, e)$ may be undefined and $\phi^i_m(\sigma, e[\sigma])$ defined). In other terms, transitions are preserved under substitutions but not necessarily reflected (as stated also by Equation 4).

4.3 Indexed labelled transition systems for the Fusion calculus

The interpretation of Fusion processes will be B_F-coalgebras, that is, an object "of states" X of $Set^{\mathbb{F}}$, together with a behaviour map $\beta : X \to B_F(X)$. $B_F(X)$ is a rather complex object (a presheaf of partial natural transformations), but we can simplify this definition, as follows:

Proposition 4.2 *The category of B_F-coalgebras is equivalent to the following category:*

- *objects: pairs $(X, (\beta^i : X \times E \rightharpoonup L(X \times E))_{i \in I})$, for X in $Set^{\mathbb{F}}$;*
- *a map $\phi : (X, \beta) \to (Y, \gamma)$ is a map $\phi : X \to Y$ such that $\phi \times id_E : X \times E \to Y \times E$ is a map of D-coalgebras.*

Proof. Follows from adjunction (8), implying that there is a 1-to-1 correspondence

$$\frac{X \longrightarrow B_F(X)}{X \times E \longrightarrow D(X \times E)} \tag{10}$$

and expanding $D = \tilde{K}L$, a map $X \times E \longrightarrow D(X \times E)$ is a finite family of partial maps $X \times E \rightharpoonup L(X \times E)$. □

In virtue of this result, we can give an explicit description of B_F-coalgebras, by means of a suitable notion of "$\mathbb{F}E$-labelled transition system" (See [5,12,8] for similar kinds of "indexed" labelled transition systems).

Definition 4.3 ($\mathbb{F}E$-LTS) *The presheaf of* actions *Act is*

$$Act \triangleq N \times N + N \times N + N + N + 1 + N \times N$$

For each $n \in \mathbb{F}$, we denote the elements of Act_n as xy, $\bar{x}y$, x, $\bar{x}$, $\mathbf{1}$, $x{=}y$, respectively.

An $\mathbb{F}E$-labelled transition system $\mathcal{L} = (X, \rightarrow)$ is a presheaf X of $Set^{\mathbb{F}}$ and a graph $\rightarrow$ such that

- *nodes are labelled with the* elements *of $X \times E$, that is, elements of $\int X \times E = \{(n, P, e) \mid n \in \mathbb{F}, P \in X_n, e \in E_n\}$;*
- *edges are labelled s.t. if $(n, P, e) \xrightarrow{\alpha} (m, Q, d)$ then $\alpha \in Act_n$ and*
  - *if $\alpha \in \{xy, \bar{x}y, \mathbf{1}, x{=}y\}$ then $m = n$*
  - *if $\alpha \in \{x, \bar{x}\}$ then $m = \delta n = n + 1$;*
- *(Closure) for $\kappa \in \{Id, \delta\}$, if $(n, P, e) \xrightarrow{\alpha} (\kappa n, Q, \kappa d)$ then for all $\sigma : n \to m$: $(n[\sigma], P[\sigma], e[\sigma]) \xrightarrow{\alpha[\sigma]} (\kappa n[\sigma], Q[\kappa\sigma], \kappa d[\sigma])$*

It is evident that in a transition $(n, P, e) \xrightarrow{\alpha} (m', Q, d)$, m' is completely determined by n and α; therefore, we will write transitions as $(P, e) \xrightarrow{\alpha}_n (Q, d)$. Thus, the

configurations of a $\mathbb{F}E$-LTS are processes together with the state of equivalence of the allocated variables, and the labels are Fusion action.

Proposition 4.4 *$\mathbb{F}E$-labelled transition systems are in 1-to-1 correspondence with B_F-coalgebras.*

Proof. In virtue of Proposition 4.2, a B_F-coalgebra is a pair $(X, (\phi^i : X \times E \rightharpoonup L(X \times E))_{i \in I})$. From this, it is easy to construct an $\mathbb{F}E$-labelled transition system $(X, \rightarrow)$, whose graph is defind as

$$(P, e) \xrightarrow{\alpha}_n (Q, d) \iff \text{for some } i \in I : \phi^i_n(P, e) = (\alpha, Q, d)$$

The "partial naturality" of each ϕ^i, as in equation (2), corresponds to condition (Closure) in Definition 4.3. The converse correspondence is also easy. □

We can estabilish a clear relation between $\mathbb{F}E$-LTS and the original semantics of Figure 2. In fact, this LTS induces a $\mathbb{F}E$-LTS on $Proc$:

Proposition 4.5 *For all n in $\mathbb{F}$, let us define $(P, e) \xrightarrow{\alpha}_n (Q, d)$ as follows:*

- *$P \in Proc_n$, $e \in E_n$, $\alpha \in Act_n$;*
- *for $\alpha \in \{xy, \bar{x}y, \mathbf{1}\}$:*

$$(P, e) \xrightarrow{\alpha}_n (Q, d) \iff Q \in Proc_n, d = e \text{ and } P[e] \xrightarrow{\alpha[e]} Q[e];$$

- *for $\alpha \in \{x, \bar{x}\}$:*

$$(P, e) \xrightarrow{\alpha}_n (Q, d) \iff Q \in Proc_{n+1}, d = e + 1 \text{ and } P[e] \xrightarrow{\alpha[e](z)} Q[e + 1]$$

  where $z \notin fv(\alpha[e])$ is the new added name;

- *$(P, e) \xrightarrow{x=y}_n (Q, d)$ iff $Q \in Proc_n, d = e \cup x{=}y$ and $P[e] \xrightarrow{(x=y)[e]} Q[e]$;*

Then, $\mathcal{L}_F \triangleq (Proc, \rightarrow)$ is a $\mathbb{F}E$-LTS.

Notice that we use equivalences e as substitutions, which is correct because an $e \in E_n$ is a coequalizer, i.e. a surjective map $e : n \rightarrow k$, for some k (equation (1)).

4.4 Indexed bisimulation

Using $\mathbb{F}E$-labelled transition systems, we can give an explicit description of coalgebraic B_F-bisimulations.

Proposition 4.6 *The following are equivalent:*

- *A B_F-bisimulation on a functor $A \in Set^{\mathbb{F}}$ (that is, a span between a B_F-coalgebra whose carrier is A, and itself);*
- *A family of symmetric relations*

$$(R_n \subseteq (A_n \times E_n) \times (A_n \times E_n))_{n \in \mathbb{N}}$$

  such that, if $(P, e) R_n (Q, d)$, then

(i) *if* $(P,e) \xrightarrow{\alpha}_n (P',e')$ *where* $\alpha \in \{xy, \bar{x}y, \mathbf{1}, x{=}y\}$, *then there exists* $(Q',d') \in A_n \times E_n$ *such that* $(Q,d) \xrightarrow{\alpha}_n (Q',d')$ *and* $(P',e')R_n(Q',d')$
(ii) *if* $(P,e) \xrightarrow{\alpha}_n (P',e')$ *where* $\alpha \in \{x, \bar{x}\}$, *then there exists* $(Q',d') \in A_{n+1} \times E_{n+1}$ *such that* $(Q,d) \xrightarrow{\alpha}_n (Q',d')$ *and* $(P',e')R_{n+1}(Q',d')$
(iii) *and moreover, for all* $\sigma : n \to m$ *in* $\mathbb{F}$: $(P[\sigma], e[\sigma])R_m(Q[\sigma], d[\sigma])$.

The characterization of B_F-bisimulations provided by Proposition 4.6 allows us to compare B_F-bisimulations with the original notions of fusion bisimulation and hyperbisimulation of Definition 2.2. The clauses in Definition 2.2 about communication actions are covered by clauses (i), (ii) in Proposition 4.6; the condition about bound actions in Definition 2.2 is represented in clause (ii) by the fact that, after a bound transition, the resulting processes are bisimilar via a different relation, on a set of names where the fresh name has been allocated.

In fact, B_F-bisimulations over $Proc$ endowed with the $\mathbb{F}E$-LTS $\mathcal{L}_F$ coincide with hyperequivalences, as shown next.

Theorem 4.7 *Two processes* $P, Q \in Proc_n$ *are hyperequivalent if and only if* $(P,\emptyset)$ *and* $(Q,\emptyset)$ *are* B_F*-bisimilar.*

Proof. ($\Rightarrow$) Let $P, Q \in Proc_n$ be two hyperequivalent processes; this means that there exists a substitution-closed fusion bisimulation $\mathcal{S}$ such that $P\mathcal{S}Q$. Then, we can define a B_F-bisimilarity $(R_n)_{n\in\mathbb{N}}$ over $\mathcal{L}_F$ as follows:

$$\text{for } n \in \mathbb{N}, S, T \in Proc_n, e, d \in E_n : \quad (S,e)R_n(T,d) \iff S[e]\ \mathcal{S}\ T[d] \tag{11}$$

It can be checked that this gives a B_F-bisimulation, using Proposition 4.6; in particular, clause (iii) holds because $\mathcal{S}$ is closed under substitutions. Obviously $(P,\emptyset)R_n(Q,\emptyset)$, hence the thesis.

($\Leftarrow$) Let $(R_n)_{n\in\mathbb{N}}$ be a B_F-bisimulation over $\mathcal{L}_F$, such that $(P,\emptyset)R_n(Q,\emptyset)$. We can define a relation $\mathcal{S}$ over processes using the same definition (11) above. Clearly $P\mathcal{S}Q$; we have to check that $\mathcal{S}$ is a hyperbisimulation. The only subtle case is that of fusion transitions. Let $T_1, T_2 \in Proc_n$ such that $T_1\mathcal{S}T_2$ and $T_1 \xrightarrow{x=y} T_1'$; then, by Proposition 4.5 this means that $(T_1,\emptyset) \xrightarrow{x=y}_n (T_1', \{x{=}y\})$ in $\mathcal{L}_F$. But $T_1\mathcal{S}T_2$ implies that $(T_1,\emptyset)R_n(T_2,\emptyset)$, hence there exists a T_2' such that $(T_2,\emptyset) \xrightarrow{x=y}_n (T_2', \{x{=}y\})$, and that $(T_1', \{x{=}y\})R_n(T_2', \{x{=}y\})$. By definition of $\mathcal{L}_F$ and $\mathcal{S}$, this implies that $T_2 \xrightarrow{x=y} T_2'$ and $T_1'\{x{=}y\}\ \mathcal{S}\ T_2'\{x{=}y\}$.

Closure under substitutions is guaranteed by Proposition 4.6(iii). □

5 Categorical rules

In this section we describe the GSOS-like[3] labelled transition system in Figure 2 as a natural transformation between suitable endofunctors on $Set^{\mathbb{F}}$, following the approach pioneered by Turi and Plotkin in [26]. The main advantage of such a presentation is that, if the behaviour functor preserves weak pullbacks, we get a compositional fully abstract semantics.

[3] These rules are not properly GSOS due to the conditions on bound names in rules *Open* and *Pass*.

We define a natural transformation $\mathcal{S}$ from $\Sigma(Id \times B)$ to BT, which are two endofunctors on $Set^{\mathbb{F}}$. (In this section we omit the index $_F$ from Σ_F, B_F). This means that we have to define a family of natural transformations

$$\mathcal{S}_X : \Sigma(X \times BX) \longrightarrow BTX \qquad \text{in } Set^{\mathbb{F}}$$

natural in $X \in Set^{\mathbb{F}}$. By exploiting the adjunction (10), we can consider instead the following simpler form

$$\mathcal{S}_X : \Sigma(X \times BX) \times E \longrightarrow D(TX \times E) \qquad \text{in } Set^{\mathbb{F}} \tag{12}$$

which makes explicit that the computation is "stateful", the state being represented by the E in the domain and the codomain.

By expanding Σ, it is easy to see that $\mathcal{S}_X$ is the product of several transformations, one for each syntactic constructor in the language:

$$\begin{aligned}
\mathcal{S}^{\mathbf{0}}_X &: E \longrightarrow D(TX \times E)\\
\mathcal{S}^{\text{in}}_X &: N \times N \times X \times BX \times E \longrightarrow D(TX \times E)\\
\mathcal{S}^{\text{out}}_X &: N \times N \times X \times BX \times E \longrightarrow D(TX \times E)\\
\mathcal{S}^{\text{par}}_X &: X \times BX \times X \times BX \times E \longrightarrow D(TX \times E)\\
\mathcal{S}^{\text{res}}_X &: N \times \delta X \times \delta BX \times E \longrightarrow D(TX \times E)
\end{aligned}$$

In turn, each of these natural transformations is defined by collecting the rules which specify the behaviour of the corresponding constructor.

Recall that $D = \tilde{K}L$; hence, according to Remark 3.5, each $\mathcal{S}^{op}_X$ is a finite family of partial natural transformations with codomain $L(TX \times E)$; for instance $\mathcal{S}^{\mathbf{0}}_X$ is a family $\big((\mathcal{S}^{\mathbf{0}}_X)^i : E \rightharpoonup L(TX \times E)\big)_{i \in I}$. For sake of simplicity, we present such a family as in Eq. 3, that is, as an $\mathbb{F}$-indexed family of finite-set valued function

$$\big((\mathcal{S}^{\mathbf{0}}_X)_n : E_n \to \wp_f(L(TX \times E)_n)\big)_{n \in \mathbb{F}}$$

subject to the "weak naturality" of Eq. 4; similarly for $\mathcal{S}^{\text{in}}_X$, $\mathcal{S}^{\text{out}}_X$, $\mathcal{S}^{\text{par}}_X$, $\mathcal{S}^{\text{res}}_X$.

There is no rule for $\mathbf{0}$, so it is simply $(\mathcal{S}^{\mathbf{0}}_X)_n(e) = \emptyset$.

For input and output we have only the $Pref$ rule, so $\mathcal{S}^{\text{in}}_X$ and $\mathcal{S}^{\text{out}}_X$ are as follows:

$$\begin{aligned}
&(\mathcal{S}^{\text{in}}_X)_n : n \times n \times X_n \times Set^{\mathbb{F}}(\mathbf{y}(n) \times E, \tilde{K}L(X \times E)) \times E_n \longrightarrow \wp_f(L(TX \times E)_n)\\
&(\mathcal{S}^{\text{in}}_X)_n(x, y, P, \beta, e) = \{(xy, P, e)\}
\end{aligned}$$

$$\begin{aligned}
&(\mathcal{S}^{\text{out}}_X)_n : n \times n \times X_n \times Set^{\mathbb{F}}(\mathbf{y}(n) \times E, \tilde{K}L(X \times E)) \times E_n \longrightarrow \wp_f(L(TX \times E)_n)\\
&(\mathcal{S}^{\text{out}}_X)_n(x, y, P, \beta, e) = \{(\bar{x}y, P, e)\}
\end{aligned}$$

Notice that at $n = \emptyset$, these functions are vacuous because the domain is empty.

We can define $\mathcal{S}^{\text{par}}_X$ as the union of six transformations, corresponding to the rules Par_l, Par_r, Com, $Close_l$, $Close_r$ and $Close$:

$$\begin{aligned}
\rho^{Par_l}, \rho^{Par_r}, \rho^{Com}, \rho^{Close_l}, \rho^{Close_r}, \rho^{Close} &: X \times BX \times X \times BX \times E\\
&\longrightarrow \tilde{K}(Act \times TX \times E)
\end{aligned}$$

$$\rho^{Par_l}_n(P,\beta,Q,\gamma,e) = \{(\alpha, P'|Q, e') \mid (\alpha,P',e') \in \beta_n(id_n,e)\}$$
$$\rho^{Par_r}_n(P,\beta,Q,\gamma,e) = \{(\alpha, P|Q', e') \mid (\alpha,Q',e') \in \gamma_n(id_n,e)\}$$
$$\begin{aligned}\rho^{Com}_n(P,\beta,Q,\gamma,e) = \{(x{=}y, P'|Q', e_1 \cup e_2 \cup \{x{=}y\}) \mid \\ \text{for some } z,w \in n \text{ such that } e(z)=e(w): \\ ((zx,P',e_1) \in \beta_n(id_n,e) \wedge (\bar{w}y,Q',e_2) \in \gamma_n(id_n,e)) \vee \\ ((\bar{z}x,P',e_1) \in \beta_n(id_n,e) \wedge (wy,Q',e_2) \in \gamma_n(id_n,e))\}\end{aligned}$$

$$\begin{aligned}\rho^{Close_l}_n(P,\beta,Q,\gamma,e) = \{(1, P'[y/x]|Q', e_{1\restriction n} \cup e_2) \mid x = new_n() \text{ and} \\ \text{for some } z,w \in n \text{ such that } e(z)=e(w): \\ ((z,P',e_1) \in \beta_n(id_n,e) \wedge (\bar{w}y,Q',e_2) \in \gamma_n(id_n,e)) \vee \\ ((\bar{z},P',e_1) \in \beta_n(id_n,e) \wedge (wy,Q',e_2) \in \gamma_n(id_n,e))\}\end{aligned}$$

$$\begin{aligned}\rho^{Close_r}_n(P,\beta,Q,\gamma,e) = \{(1, P'|Q'[x/y], e_1 \cup e_{2\restriction n}) \mid y = new_n() \text{ and} \\ \text{for some } z,w \in n \text{ such that } e(z)=e(w): \\ ((zx,P',e_1,\sigma) \in \beta_n(id_n,e) \wedge (\bar{w},Q',e_2,\sigma) \in \gamma_n(id_n,e)) \vee \\ ((\bar{z}x,P',e_1,\sigma) \in \beta_n(id_n,e) \wedge (w,Q',e_2,\sigma) \in \gamma_n(id_n,e))\}\end{aligned}$$

$$\begin{aligned}\rho^{Close}_n(P,\beta,Q,\gamma,e) = \{(1, (x)(P'|Q'), e_{1\restriction n} \cup e_{2\restriction n}) \mid x = new_n() \text{ and} \\ \text{for some } z,w \in n \text{ such that } e(z)=e(w): \\ ((z,P',e_1) \in \beta_n(id_n,e) \wedge (\bar{w},Q',e_2) \in \gamma_n(id_n,e)) \vee \\ ((\bar{z},P',e_1) \in \beta_n(id_n,e) \wedge (w,Q',e_2) \in \gamma_n(id_n,e))\}\end{aligned}$$

Notice that in ρ^{Com}, the equivalences in the transitions contain the effects from both subprocesses, and also the equivalence caused by the communication. In the "close" rules, instead, the resulting equivalences do not have fusions caused by the communication, because this acts either as an immediate substitution (in left and right close), or as the (silent) unification of the hidden names (in the last rule). The local name x introduced by bound transitions is removed from the equivalence by the restrictions $e_{1\restriction n}, e_{2\restriction n}$ along the inclusion $n \hookrightarrow n+1$.

Finally, $\mathcal{S}^{\text{res}}_X$ is defined in terms of rules *Open*, *Scope* and *Pass*; in fact it can be obtained from the union of the following three transformations:

$$\rho^{Open} : \delta X \times \delta BX \times E \longrightarrow \tilde{K}(Act \times \delta(TX \times E))$$
$$\rho^{Open}_n : X_{n+1} \times Set^{\mathbb{F}}(\mathbf{y}(n{+}1) \times E, \tilde{K}L(X{\times}E)) \times E_n \longrightarrow \wp_f(Act_n \times (TX)_{n+1} \times E_{n+1})$$
$$\begin{aligned}\rho^{Open}_n(P,\beta,e) = \{(y, P', e') \mid y \in n, x = new_n(), (yx,P',e') \in \beta_{n+1}(id_{n+1}, e{+}1)\} \\ \cup \{(\bar{y}, P', e') \mid y \in n, x = new_n(), (\bar{y}x,P',e') \in \beta_{n+1}(id_{n+1}, e{+}1)\}\end{aligned}$$

$$\rho^{Pass}, \rho^{Scope} : \delta X \times \delta BX \times E \longrightarrow \tilde{K}(Act \times TX \times E)$$
$$\rho^{Pass}_n(P,\beta,e) = \{(\alpha, (x)P', e'_{\restriction n}) \mid x = new_n(), (\alpha,P',e') \in \beta_{n+1}(id_{n+1}, e+1)\}$$
$$\begin{aligned}\rho^{Scope}_n(P,\beta,e) = \{(1, P'[y/x], e'[y/x]) \mid y \in n, x = new_n(), \\ (x{=}y, P', e') \in \beta_{n+1}(id_{n+1}, e+1)\}\end{aligned}$$

In ρ^{Scope}, notice that the process and the state are changed by the substitution $[y/x] : n+1 \to n$, which replaces the fresh variable x with y.

Checking naturality (as per Eq. 4) of these definition is a tedious but easy task.

The most important consequence of this definition is the following result.

Theorem 5.1 *There exists a (unique) semantics* $[\![\_]\!] : Proc \longrightarrow \nu B_F$ *in* $Set^{\mathbb{F}}$ *which is both compositional and fully abstract, that is, for all processes* $P, Q \in Proc_n$*:* $[\![P]\!]_n = [\![Q]\!]_n$ *iff* P *and* Q *are* B_F*-bisimilar (and hence hyperequivalent).*

Proof. Follows from general results in [26,6] about bialgebras with distributive laws: the semantics exists because $Proc$ can be endowed with a B_F-coalgebraic structure, and νB_F with a Σ_F-algebraic structure. As a consequence, $[\![\_]\!]$ is both the unique map from the initial Σ_F-algebra (i.e., $Proc$) and the unique map to the final B_F-coalgebra, which exists in virtue of Proposition 3.6. However, in order to apply this theory, we have to prove that B_F on $Set^{\mathbb{F}}$ preserves weak pullbacks.

Recall that $B_F = (\tilde{K}L(\_ \times E))^E$ and that $L(\_ \times E)$ is defined by a polynomial containing some δ's (equation 7). It is well known that in presheaf categories, polynomial functors preserve weak pullbacks; the same for δ [7]. $\tilde{K}$ preserves weak pullbacks by Proposition 3.4. It remains to prove that $(\_)^E$ preserves weak pullbacks. As all limits in presheaf categories, pullbacks are computed pointwise; at n, it is $((\_)^E)_n = Set^{\mathbb{F}}(\mathbf{y}(n) \times E, \_)$, and for any A, the hom-functor $Set^{\mathbb{F}}(A, \_) : Set^{\mathbb{F}} \to Set$ preserves limits. □

6 Conclusions

In this paper we have provided a categorical account of the Fusion calculus. First, we have presented a new, congruence-free labelled transition system for the Fusion calculus. Then, we have described the syntax of Fusion as the initial algebra of a suitable signature monad, and the semantics of processes as coalgebras of a behaviour "stateful" endofunctor, over the category of presheaves over finite sets. An explicit presentation of these coalgebras has been given, as "stateful" $\mathbb{F}E$*-labelled transition systems*. The coalgebraic bisimulations on these systems turns out to be closed under substitutions (hence a congruence), and corresponding to hyperequivalence. Then, we have given a bialgebraic semantics for the Fusion calculus, by modelling the rules of the new labelled transition system as an abstract categorical rule between endofunctors over $Set^{\mathbb{F}}$. As a consequence, we have got a semantics which is both compositional and fully abstract, that is, two processes have the same semantics iff they are coalgebraically bisimilar, and hence hyperequivalent.

The categorical presentation of the Fusion calculus we have given, sheds also some light on the "name equivalence state" which is hidden inside Fusion processes. In fact, the nodes of a $\mathbb{F}E$-labelled transition systems are pairs (P, e), to indicate that a system configuration is not simply a process, but a process together with the state of the equivalence over its names. This is actually confirmed in some offsprings of Fusion, such as the Explicit Fusions Calculus [27] and CC-Pi [4]; in both these calculi, processes have an explicit and clearly identified part which carries the information about the state of name/term equivalence. The same happens in the bigraphical representations of Fusion processes given in [13]: a specific part of the bigraph representing an agent, encodes the equivalence classes of that agent. It is interesting future work to investigate whether the model and techniques presented in this paper can be applied to model also these similar calculi, and also Fu's χ-calculus

[11], a concurrent computation model similar to Fusion, and the open π-calculus [22]; in both these calculi, bisimulation is closed under substitution.

Presheaf categories have been widely used for modelling languages with dynamically allocable entities, such as (bound) variables, (fresh) names, reference, etc; see e.g. [23,7,14,6,12,21,18,8]. Maybe the work closest to ours is the treatment of π-calculus in [6]: also in that case, the right behavior functor over $Set^{\mathbb{F}}$ is obtained by lifting the "intuitive" endofunctor given over another category, along an adjunction (although not the same as the one considered in this paper).

Finally, we mention that Buscemi and Montanari have given a compositional coalgebraic model of (a fragment of) the Fusion calculus [3]. Their techniques are more set-theoretic, using a first-order approach with a set of names endowed with "permutation" and "shift" and operators, thus akin de Bruijn indexes. We can envisage a connection between their approach and the model presented in this paper, which we leave as future work.

Acknowledgments The author thanks Sam Staton and the anonymous referees, for useful and important comments on the preliminary version of this work.

References

[1] J. Adamek, editor. *Coalgebraic Methods in Computer Science*, volume 106 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.

[2] J. Adámek. Introduction to coalgebra. *Theory and Applications of Categories*, 14(8):157–199, 2005.

[3] M. G. Buscemi and U. Montanari. A compositional coalgebraic model of a fragment of fusion calculus. *Electr. Notes Theor. Comput. Sci.*, 162:135–139, 2006.

[4] M. G. Buscemi and U. Montanari. Cc-pi: A constraint-based language for specifying service level agreements. In R. D. Nicola, editor, *Proc. ESOP*, volume 4421 of *Lecture Notes in Computer Science*, pages 18–32. Springer, 2007.

[5] G. L. Cattani and P. Sewell. Models for name-passing processes: interleaving and causal. *Inf. Comput.*, 190(2):136–178, 2004.

[6] M. Fiore and D. Turi. Semantics of name and value passing. In H. Mairson, editor, *Proc. 16th LICS*, pages 93–104, Boston, USA, 2001. IEEE Computer Society Press.

[7] M. P. Fiore, G. D. Plotkin, and D. Turi. Abstract syntax and variable binding. In Longo [17], pages 193–202.

[8] M. P. Fiore and S. Staton. Comparing operational models of name-passing process calculi. *Inf. Comput.*, 204(4):524–560, 2006.

[9] C. Fournet, F. L. Fessant, L. Maranget, and A. Schmitt. Jocaml: A language for concurrent distributed and mobile programming. In J. Jeuring and S. L. P. Jones, editors, *Advanced Functional Programming*, volume 2638 of *Lecture Notes in Computer Science*, pages 129–158. Springer, 2002.

[10] P. Freyd. Numerology in topoi. *Theory and Applications of Categories*, 16(19):522–528, 2006.

[11] Y. Fu. Variations on mobile processes. *Theor. Comput. Sci.*, 221(1-2):327–368, 1999.

[12] N. Ghani, K. Yemane, and B. Victor. Relationally staged computation in calculi of mobile processes. In Adamek [1].

[13] D. Grohmann and M. Miculan. Reactive systems over directed bigraphs. In L. Caires and V. Vasconcelos, editors, *Proc. CONCUR 2007*, volume 4703 of *Lecture Notes in Computer Science*, pages 380–394. Springer-Verlag, 2007.

[14] M. Hofmann. Semantical analysis of higher-order abstract syntax. In Longo [17], pages 204–213.

[15] P. Johnstone, J. Power, T. Tsujishita, H. Watanabe, and J. Worrell. On the structure of categories of coalgebras. *Theor. Comput. Sci.*, 260(1-2):87–117, 2001.

[16] P. T. Johnstone. *Sketches of an Elephant (Volume 2)*. Number 44 in Oxford Logic Guides. Oxford University Press, New York, 2002.

[17] G. Longo, editor. *Proceedings, Fourteenth Annual Symposium on Logic in Computer Science*, Trento, Italy, 1999. IEEE Computer Society Press.

[18] M. Miculan and K. Yemane. A unifying model of variables and names. In V. Sassone, editor, *Proc. FOSSACS'05*, volume 3441 of *Lecture Notes in Computer Science*, Apr. 2005.

[19] E. Moggi. Notions of computation and monads. *Information and Computation*, 1, 1993.

[20] J. Parrow and B. Victor. The Fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS '98*, pages 176–185. IEEE Computer Society Press, July 1998.

[21] J. Power and M. Tanaka. Binding signatures for generic contexts. In P. Urzyczyn, editor, *TLCA*, volume 3461 of *Lecture Notes in Computer Science*, pages 308–323. Springer, 2005.

[22] D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996.

[23] I. Stark. A fully abstract domain model for the π-calculus. In *Proc. LICS'96*, pages 36–42. IEEE, 1996.

[24] S. Staton. *Name-passing process calculi: operational models and structural operational semantics*. PhD thesis, Computer Laboratory, University of Cambridge, 2007.

[25] S. Staton. Personal communication, 2008.

[26] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. 12$^{\text{th}}$ LICS Conf.*, pages 280–291. IEEE Computer Society Press, 1997.

[27] L. Wischik and P. Gardner. Explicit Fusions. *Theor. Comput. Sci.*, 340(3):606–630, 2005.