UNIVERSITÀ DI PISA
DIPARTIMENTO DI INFORMATICA

TECHNICAL REPORT: TR-05-17

A Compositional Coalgebraic Model of Monadic Fusion Calculus

Maria Grazia Buscemi Ugo Montanari

June 20, 2005

ADDRESS: via F. Buonarroti 2, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

A Compositional Coalgebraic Model of Monadic Fusion Calculus

Maria Grazia Buscemi and Ugo Montanari

Dipartimento di Informatica, Università di Pisa, Italy.
{buscemi,ugo}@di.unipi.it

Abstract. We propose a compositional coalgebraic semantics of the Fusion calculus of Parrow and Victor in the version with explicit fusions by Gardner and Wischik. We follow a recent approach developed by the same authors and previously applied to the pi-calculus for lifting calculi with structural axioms to bialgebraic models. In our model, the unique morphism to the final bialgebra induces a bisimilarity relation which coincides with hyperequivalence and which is a congruence with respect to the operations. Interesting enough, the explicit fusion approach allows to exploit for the Fusion calculus essentially the same algebraic structure used for the pi-calculus.

1 Introduction

Fusion calculus has been introduced as a variant of the pi-calculus [9]. It makes input and output operations fully symmetric and enables a more general name matching mechanism during synchronisation. A fusion is a name equivalence that allows to use interchangeably in a term all names of an equivalence class. Computationally, a fusion is generated as a result of a synchronisation between two complementary actions, and it is propagated to processes running in parallel with the active one. Fusions are ideal for representing, e.g., forwarders for objects that migrate among locations [7], or forms of pattern matching between pairs of messages [8].

In Fusion calculus, a fusion, as soon as it is generated, is immediately applied to the whole system and has the effect of a (possibly non-injective) name substitution. On the other hand, the version of the calculus with explicit fusions [6, 7] aims at propagating fusions to the environment in an asynchronous way. Explicit fusions are processes that exist concurrently with the rest of the system and enable to freely use two names one for the other.

Structural operational semantics [1, 13] is a well-established technique to provide process calculi and specification languages with an interpretation. A transition system is inductively derived from a set of transition rules that describe the behaviour of every construct of the language.

A coalgebraic framework [14] presents several advantages: morphisms between coalgebras (cohomomorphisms) enjoy the property of "reflecting behaviours" and thus they allow, for example, to characterise bisimulation equivalences as kernels of morphisms and bisimilarity as the kernel of the morphism to the final coalgebra. Also adequate temporal logics and proof methods by coinduction fit nicely into the picture.

However, in the ordinary coalgebraic framework, the states of transition systems are seen simply as set elements, i.e. the algebraic structure needed for composing programs and states is disregarded. Bialgebraic models take a step forward in this direction: they aim at capturing interactive systems which are compositional. Roughly, bialgebras [16] are structures that can be regarded as coalgebras on a category of algebras rather than on the category **Set**, or, symmetrically, as algebras on a category of coalgebras. For them bisimilarity is a congruence, namely compositionality of abstract semantics is automatically guaranteed.

When considering mobile interactive systems, like the pi-calculus, the ordinary coalgebraic approach cannot be directly applied, since the generation of new names requires special conditions on the inference rules and on the definition of bisimulations. The bialgebraic approach, instead, fits well: it is enough to consider the states as forming an algebra of name permutations [10, 11]. However, the interaction of structural axioms with inference rules makes the application of the bialgebraic approach problematic, if more complex operations are taken into account. To overcome this difficulty, in [2] it has been proved that calculi defined by De Simone inference rules and equipped with structural axioms can be lifted to bialgebras, provided that axioms bisimulate. In the same paper, the approach has been applied to a version of pi-calculus.

In this paper we apply the general theory presented in [2] to the fusion calculus of Parrow and Victor, in order to provide a bialgebraic model of the calculus. We argue that this result does not only concern the fusion calculus but it could fit within theoretical foundations of languages based on pattern matching.

Since bisimilarity in the π-calculus fails to be a congruence due to input prefix, the model in [2] is compositional only with respect to parallel composition and restriction; constants are introduced to map π-agents whose out-most operator is neither parallel composition nor restriction. Moreover, the theory in *loc. cit.* does not apply to late and open π-calculus as this would require the introduction of arbitrary (possibly non-injective) name substitutions. Our present model of the fusion calculus, instead, is fully compositional with respect to the operations of the calculus. This is accomplished by the introduction of explicit fusions into the underlying algebra. Indeed, the combination of explicit fusions and restriction allows to derive a name substitution operator which behaves like the standard capture-avoiding substitution.

We introduce a permutation algebra enriched with the operations of the calculus plus constants modelling explicit fusions. We then prove that the conditions required by [2] are satisfied. Remarkably enough, explicit fusions enable us to model substitutions within our theory, while keeping essentially the same permutation algebra considered in [2] for the pi-calculus. No non-injective substitution operations are introduced in the algebra: rather, their observable effects are simulated by De Simone inference rules which saturate process behaviours, while still keeping the nice property of asynchronous propagation typical of explicit fusions. We claim that the translation of fusion agents in our algebra is fully abstract with respect to Parrow and Victor hyperequivalence. As in [18], closure with respect to substitution is obtained by adding in parallel at each step any possible fusion.

Structure of the paper Section 2 contains the background on permutations, fusion calculus, and theory of bialgebras. In Section 3 we define a permutation algebra for the fusion calculus, along with a transition system with an algebraic structure *lts* and we prove that it can be lifted to a bialgebra. Moreover, we prove that fusion agents can be translated into terms of our algebra and that such a translation is fully abstract with respect to fusion hyperequivalence. The complete proof is reported in the appendix. Finally, Section 4 contains some concluding remarks and directions for future work.

2 Background

2.1 Names, Fusion and Permutations

We need some basic definitions and properties on names, fusions and permutations of names. We denote with $\mathfrak{N} = \{x_0, x_1, x_2, \ldots\}$ the infinite, countable, totally ordered set of *names* and we use $x, y, z \ldots$ to denote names.

Name fusions (or, simply, *fusions*) are total equivalence relations on $\mathfrak{N}$ with only finitely many non-singular equivalence classes. Fusions are ranged over by $\varphi, \psi, \ldots$. We let:

- $\mathrm{n}(\varphi)$ denote $\{x : x\varphi y \text{ for some } y \neq x\}$;
- τ denote the identity fusion (i.e., $\mathrm{n}(\tau) = \emptyset$);
- $\varphi + \psi$ denote the finest fusion which is coarser than φ and ψ, that is $(\varphi \cup \psi)^\star$;
- φ_{-z} denote $\varphi - (\{z\} \times \mathfrak{N} \cup \mathfrak{N} \times \{z\}) \cup \{(z,z)\}$;
- $\varphi[x]$ denote the equivalence class of x in φ;
- $\varphi \sqsubseteq \psi$ denote that φ is finer that ψ, i.e., for all $x \in \mathfrak{N}$, $\varphi[x] \subseteq \psi[x]$;
- $\{x = y\}$ denote $\{(x,y),(y,x)\}^\star$.

A *name substitution* is a function $\sigma : \mathfrak{N} \to \mathfrak{N}$. We denote with $\sigma \circ \sigma'$ the composition of substitutions σ and σ'; that is, $\sigma \circ \sigma'(x) = \sigma(\sigma'(x))$. We use σ to range over substitution and we denote with $[y_1 \mapsto x_1, \cdots, y_n \mapsto x_n]$ the substitution that maps x_i into y_i for $i = 1, \ldots, n$ and which is the identity on the other names. The *identity substitution* is denoted by id.

A substitution σ *agrees with* a fusion φ if $\forall x, y : x\varphi y \Leftrightarrow \sigma(x) = \sigma(y)$. A *substitutive effect* of a fusion φ is a substitution σ agreeing with φ such that $\forall x, y : \sigma(x) = y \Rightarrow x\varphi y$ (i.e., σ sends all members of the equivalence class to one representative of the class).

Name permutations, ranged over by ρ, are bijective name substitutions. We abbreviate by $[y \leftrightarrow x]$ the permutation $[y \mapsto x, x \mapsto y]$. Given a permutation ρ, we define permutation ρ_{+1} as follows:

$$\frac{-}{\rho_{+1}(x_0) = x_0} \qquad\qquad \frac{\rho(x_n) = x_m}{\rho_{+1}(x_{n+1}) = x_{m+1}}$$

Essentially, permutation ρ_{+1} is obtained from ρ by shifting its correspondences to the right by one position.

2.2 The Fusion Calculus

In this section we give an overview of the fusion calculus, which has been introduced in [12]. Here we consider a *monadic* version of the calculus.

The fusion calculus *agents*, ranged over by $P, Q, \ldots$, are closed (wrt. variables X) terms defined by the syntax:

$$P ::= \mathbf{0} \;\big|\; \pi.P \;\big|\; P|P \;\big|\; (x)\,P \;\big|\; \mathtt{rec}X.P \;\big|\; X$$

where recursion is guarded, and *prefixes*, ranged over by π, are I/O actions or fusions:

$$\pi ::= \;\big|\; \bar{x}y \;\big|\; xy \;\big|\; \varphi.$$

The occurrences of x in $(x)\,P$ are bound and fusion effects with respect to x are limited to P; *free names* and *bound names* of agent P are defined as usual and we denote them with $\mathrm{fn}(P)$ and $\mathrm{bn}(P)$, respectively. Also, we denote with $\mathrm{n}(P)$ and $\mathrm{n}(\pi)$ the sets of (free and bound) names of agent term P and prefix π, respectively. We denote by $\mathcal{F}$ the set of all fusion agents.

The structural congruence, $\equiv$, between agents is the least congruence satisfying the following axioms:

(fus)	$\varphi.P \equiv \varphi.\sigma_\varphi(P)$ for σ a substitutive effect of φ
(par)	$P|\mathbf{0} \equiv P \qquad P|Q \equiv Q|P \qquad P|(Q|R) \equiv (P|Q)|R$
(res)	$(x)\,\mathbf{0} \equiv \mathbf{0} \quad (x)\,(y)\,P \equiv (y)\,(x)\,P \qquad (x)\,(P+Q) \equiv (x)\,P + (x)\,Q$
(scope)	$P|(z)\,Q \equiv (z)\,(P|Q)$ where $z \notin \mathrm{fn}(P)$

The *actions* an agent can perform, ranged over by γ, are defined by the following syntax:

$$\gamma ::= xy \;\big|\; x(z) \;\big|\; \bar{x}y \;\big|\; \bar{x}(z) \;\big|\; \varphi$$

and are called respectively *free input*, *bound input*, *free output*, *bound output* actions and *fusions*. Names x and y are free in γ ($\mathrm{fn}(\gamma)$), whereas z is a bound name ($\mathrm{bn}(\gamma)$); moreover $\mathrm{n}(\gamma) = \mathrm{fn}(\gamma) \cup \mathrm{bn}(\gamma)$. The notion of substitutive effect is extended to actions by stating that the only substitutive effect of $\gamma \neq \varphi$ is id.

The family of transitions $P \xmapsto{\gamma} Q$ is the least family satisfying the laws in Table 1. The crucial difference between the pi-calculus and the fusion calculus shows up in synchronisations: in the fusion calculus, the effect of a synchronisation is not necessarily local, and is regulated by the scope of the binder (x).

Example 1. The interaction between the agents $\overline{u}v.P$ and $ux.Q$ results in a fusion of v and x. This fusion also affect any further process R running in parallel, as illustrated below:

$$R\,|\,\overline{u}v.P\,|\,ux.Q \xmapsto{\{x=v\}} R\,|\,P\,|\,Q.$$

The binding operator (x) can be used to limit the scope of the fusion, that is:

$$R\,|\,(x)\,\big(\overline{u}v.P\,|\,ux.Q\big) \xmapsto{\tau} R\,|\,[{}^{v}\!/\!_{x}](P\,|\,Q).$$

Definition 1 (fusion bisimilarity). *A* fusion bisimulation *is a binary symmetric relation $\mathcal{S}$ between fusion agents such that $P\ \mathcal{S}\ Q$ implies:*

> *If $P \xmapsto{\gamma} P'$ with $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ then $Q \xmapsto{\gamma} Q'$ and $\sigma(P')\ \mathcal{S}\ \sigma(Q')$ for some substitutive effect σ of γ.*

P is bisimilar to Q, written $P \dot{\sim} Q$, if $P\ \mathcal{S}\ Q$ for some fusion bisimulation $\mathcal{S}$.

(F-PRE)	$\pi.P \xmapsto{\pi} P$	(F-PAR)	$\dfrac{P \xmapsto{\gamma} Q}{P \mid R \xmapsto{\gamma} Q \mid R}$ if $\mathrm{bn}(\gamma) \cap \mathrm{fn}(R) = \emptyset$
(F-COM)	$\dfrac{P \xmapsto{\bar{x}y} P' \quad Q \xmapsto{xz} Q'}{P \mid Q \xmapsto{\{y=z\}} P' \mid Q'}$	(F-SCOPE)	$\dfrac{P \xmapsto{\varphi} Q \quad z\varphi x, z \neq x}{(z)\,P \xmapsto{\varphi_{-z}} [x/z]Q}$
(F-OPEN)	$\dfrac{P \xmapsto{az} Q \quad a \notin \{z, \bar{z}\}}{(z)\,P \xmapsto{a(z)} Q}$	(F-PASS)	$\dfrac{P \xmapsto{\gamma} P'}{(z)\,P \xmapsto{\gamma} (z)\,P'}$ $z \notin \mathrm{n}(\gamma)$
(F-REC)	$\dfrac{P[\mathtt{rec}\,X.P/X] \xmapsto{\gamma} Q}{\mathtt{rec}\,X.P \xmapsto{\gamma} Q}$	(F-CONG)	$\dfrac{P \equiv P' \quad P' \xmapsto{\gamma} Q' \quad Q' \equiv Q}{P \xmapsto{\gamma} Q}$

Table 1. LTS for Fusion

Definition 2 (hyperequivalence). *A* hyperbisimulation *is a substitution closed fusion bisimulation, i.e., a fusion bisimulation $\mathcal{S}$ with the property that $P \mathcal{S} Q$ implies $\sigma(P) \mathcal{S} \sigma(Q)$ for any substitution σ. Two agents P and Q are* hyperequivalent*, written $P \sim_{he} Q$, if they are related by a hyperbisimulation.*

2.3 Bialgebras

In this section we present the relevant definitions and results about coalgebras and bialgebras. We start reviewing notions about algebras and algebraic specifications.

We recall that, for Σ a signature, a Σ-algebra $A = \langle |A|, (op^A)_{op \in \Sigma} \rangle$ consists of a carrier set $|A|$ and a family of operations such that $op^A : |A|^n \to |A|$ if $op \in \Sigma$ of arity n. We assume to have a countable set X of the variables that can be used in the terms of the algebra. A Σ-homomorphism (or simply a morphism) between two Σ-algebras A and B is a function $h : |A| \to |B|$ that commutes with all the operations in Σ, namely, for each operator $op \in \Sigma$ of arity n, we have $op^B(h(a_1), \cdots, h(a_n)) = h(op^A(a_1, \ldots, a_n))$. We denote by $\mathbf{Alg}(\Sigma)$ the category of Σ-*algebras* and Σ-*morphisms*. A Σ-algebra A satisfies an algebraic specification $\Gamma = \langle \Sigma, E \rangle$, if A satisfies all axioms in E. In this case, A is called a Γ-algebra. The category of Γ-algebras and homomorphisms is the full subcategory $\mathbf{Alg}(\Gamma) \subseteq \mathbf{Alg}(\Sigma)$.

The basic idea behind SOS specifications is to specify a transition relation by induction over the structure of the system's states. In order to make explicit this structure, rather than ordinary labelled transition systems we consider transition systems whose sets of states have an algebraic structure.

Definition 3 (transition systems). *Let $\Gamma = \langle \Sigma, E \rangle$ be an algebraic specification, and L be a set of labels. A* labelled transition system (transition system, *in brief*) *over Γ and L is a pair $lts = \langle A, \longrightarrow_{lts} \rangle$ where A is a nonempty Γ-algebra and $\longrightarrow_{lts} \subseteq |A| \times L \times |A|$ is a labelled transition relation. For $\langle p, l, q \rangle \in \longrightarrow_{lts}$ we write $p \xrightarrow{l}_{lts} q$.*

Let $lts = \langle A, \longrightarrow_{lts} \rangle$ and $lts' = \langle B, \longrightarrow_{lts'} \rangle$ be two transition systems. A morphism $h : lts \to lts'$ of transition systems over Γ and L (lts morphism, in brief) is a Γ-morphism $h : A \to B$ such that $p \xrightarrow{l}_{lts} q$ implies $f(p) \xrightarrow{l}_{lts'} f(q)$.

The notion of bisimulation and congruence on transition systems with an algebraic structure are the classical one.

Given an algebraic specification $\Gamma = \langle \Sigma, E \rangle$ and a set of labels L, a collection of SOS rules can be regarded as a specification of those transition systems over Γ and L that have a transition relation closed under the given rules.

Definition 4 (SOS rules). *Given an algebraic specification $\Gamma = \langle \Sigma, E \rangle$ and a set of labels L, a sequent $p \xrightarrow{l} q$ (over L and Γ) is a triple where $l \in L$ is a label and p, q are Σ-terms with variables in a given set X. An* SOS rule *r over Γ and L takes the form:*

$$\frac{p_1 \xrightarrow{l_1} q_1 \ \cdots \ p_n \xrightarrow{l_n} q_n}{p \xrightarrow{l} q}$$

where $p_i \xrightarrow{l_i} q_i$ for all $i = 1, \cdots n$ as well as $p \xrightarrow{l} q$ are sequents.

We say that transition system $lts = \langle A, \longrightarrow_{lts} \rangle$ satisfies *a rule r like above if each assignment to the variables in X that is a solution*[1] *to $p_i \xrightarrow{l_i} q_i$ for $i = 1, \ldots, n$ is also a solution to $p \xrightarrow{l} q$.*

Definition 5 (transition specifications). *A* transition specification *is a tuple $\Delta = \langle \Gamma, L, R \rangle$ consisting of an algebraic specification Γ, a set of labels L, and a set of SOS rules R over Γ and L. We abbreviate $\Delta = \langle \Gamma = \langle \Sigma, \emptyset \rangle, L, R \rangle$ by $\Delta = \langle \Sigma, L, R \rangle$*

A transition system over Δ is a transition system over Γ and L that satisfies rules R.

It is well known that ordinary labelled transition systems (i.e., transition systems whose states do not have an algebraic structure) can be represented as coalgebras for a suitable functor [14].

Definition 6 (coalgebras). *Let $F : \mathcal{C} \to \mathcal{C}$ be a functor on a category $\mathcal{C}$. A* coalgebra for *F, or F*-coalgebra*, is a pair $\langle A, f \rangle$ where A is an object and $f : A \to F(A)$ is an arrow of $\mathcal{C}$. A F*-cohomomorphism *(or simply F*-morphism*) $h : \langle A, f \rangle \to \langle B, g \rangle$ is an arrow $h : A \to B$ of $\mathcal{C}$ such that*

$$h; g = f; F(h). \tag{1}$$

We denote with $\mathbf{Coalg}(F)$ the category of F-coalgebras and F-morphisms.

Proposition 1. *For a fixed set of labels L, let $P_L : \mathbf{Set} \to \mathbf{Set}$ be the functor defined on objects as $P_L(X) = \mathcal{P}(L \times X + X)$, where $\mathcal{P}$ denotes the countable powerset functor, and on arrows as $P_L(h)(S) = \{\langle l, h(p) \rangle \mid \langle l, p \rangle \in S \cap L \times X\} \cup \{h(p) \mid p \in S \cap X\}$, for $h : X \to Y$ and $S \subseteq L \times X + X$. Then P_L-coalgebras are in a one-to-one correspondence with transition systems*[2] *on L, given by $f_{lts}(p) = \{\langle l, q \rangle \mid p \xrightarrow{l}_{lts} q\} \cup \{p\}$ and, conversely, by $p \xrightarrow{l}_{lts_f} q$ if and only if $\langle l, q \rangle \in f(p)$.*

[1] Given $h : X \to A$ and its extension $\hat{h} : T_{(\Sigma,E)}(X) \to A$, h is a solution to $p \xrightarrow{l} q$ for lts if and only if $\hat{h}(p) \xrightarrow{l}_{lts} \hat{h}(q)$.

[2] Notice that this correspondence is well defined also for transition systems with sets of states, rather than with algebras of states as required in Definition 3.

Definition 7 (De Simone format). *Given an algebraic specification* $\Gamma = \langle \Sigma, E \rangle$ *and a set of labels* L*, a rule* r *over* Δ *and* L *is in* De Simone format *if it has the form:*

$$\frac{\{x_i \xrightarrow{l_i} y_i \mid i \in I\}}{op(x_1, \ldots, x_n) \xrightarrow{l} p}$$

where $op \in \Sigma$*,* $I \subseteq \{1, \ldots, n\}$*,* p *is linear and the variables* y_i *occurring in* p *are distinct from variables* x_i*, except for* $y_i = x_i$ *if* $i \notin I$*.*

A De Simone proof *of a sequent* $s \xrightarrow{l} t$ *from premises* $\{x_i \xrightarrow{l_i} y_i\}_{i \in I}$ *is a proof of* $s \xrightarrow{l} t$ *from* $\{x_i \xrightarrow{l_i} y_i\}_{i \in I}$ *that is obtained using only De Simone rules in* R *and without using axioms in* E*.*

The following results are due to Turi and Plotkin ([16]) and concern *bialgebras*, that is, coalgebras in $\mathbf{Alg}(\Sigma)$. As noted in the introduction, bialgebras enjoy the property that the unique morphism to the final bialgebra, which exists under reasonable conditions, induces a bisimulation that is a congruence with respect to the operations. For our needs, it will be enough to recall a simplified theory with De Simone rules, rather than in GSOS format.

Proposition 2 (lifting of P_L**).** *Let* $\Delta = \langle \Sigma, L, R \rangle$ *be a transition specification with De Simone rules. Define* $P_\Delta : \mathbf{Alg}(\Sigma) \to \mathbf{Alg}(\Sigma)$ *as follows:*

- $|P_\Delta(A)| = P_L(|A|)$*;*
- *whenever* $\dfrac{\{x_i \xrightarrow{l_i} y_i \mid i \in I\}}{op(x_1, \ldots, x_n) \xrightarrow{l} p} \in R$ *then*

  $$\frac{\langle l_i, p_i \rangle \in S_i,\ i \in I \quad q_j \in S_j,\ j \notin I}{\langle l, p[p_i/y_i,\ i \in I,\ q_j/y_j,\ j \notin I] \rangle \in op^{P_\Delta(A)}(S_1, \ldots, S_n)};$$
- *if* $h : A \to B$ *is a morphism in* $\mathbf{Alg}(\Sigma)$ *then* $P_\Delta(h) : P_\Delta(A) \to P_\Delta(B)$ *and* $P_\Delta(h)(S) = \{\, \langle l, h(p) \rangle \mid \langle l, p \rangle \in S \cap (L \times |A|) \,\} \cup \{\, h(p) \mid p \in S \cap |A| \,\}$.

Then P_Δ *is a well-defined functor on* $\mathbf{Alg}(\Sigma)$*.*

Corollary 1. *Let* $\Delta = \langle \Sigma, L, R \rangle$ *be a transition specification with rules* R *in De Simone format.*

Any morphism $h : f \to g$ *in* $\mathbf{Coalg}(P_\Delta)$ *entails a bisimulation* $\sim_h$ *on* lts_f*, that coincides with the kernel of the morphism. Bisimulation* $\sim_h$ *is a congruence for the operations of the algebra.*

Moreover, the category $\mathbf{Coalg}(P_\Delta)$ *has a final object. Finally, the kernel of the unique* P_Δ*-morphism from* f *to the final object of* $\mathbf{Coalg}(P_\Delta)$ *is a relation on the states of* f *which coincides with bisimilarity on* lts_f *and is a congruence.*

Note that the above corollary assumes that a P_L-coalgebra can be lifted to a P_Δ-coalgebra. Indeed, such assumption is obvious in the particular case of $f : A \to P_\Delta(A)$, with $A = T_\Sigma$ and f unique by initiality, namely when A has no structural axioms and no additional constants, and lts_f is the minimal transition system satisfying Δ. The following result is due to [2] and generalise the theory described so far to transition systems with structural axioms.

Theorem 1. *Let $\mathcal{B}$ be the class of coalgebras g in* **Set** *with the following properties:*

1. $g : |B| \to P_L(|B|)$*, with* $B = T_{(\Sigma,E)}$.
2. lts_g *satisfies transition specification* $\Delta = (\langle \Sigma, E\rangle, L, R)$*, with R in De Simone format.*

Then, there is an initial coalgebra $\hat{g}$ in $\mathcal{B}$, such that $\forall g \in \mathcal{B}$, $\forall p \in B$, $p \xrightarrow{l}_{\hat{g}} q$ implies $p \xrightarrow{l}_{g} q$.

Furthermore, the transitions of $\hat{g}$ can be derived using the rules R and the following additional rule:

$$\text{(STRUCT)}\ \frac{t_1 =_E t_1' \quad t_1' \xrightarrow{l}_{\hat{g}} t_2' \quad t_2' =_E t_2}{t_1 \xrightarrow{l}_{\hat{g}} t_2}$$

where terms t_1, t_1', t_2, t_2' are in T_Σ.

Theorem 2. *Let g be the initial coalgebra in $\mathcal{B}$ as specified by Theorem 1, and let*

– $A = T_\Sigma$ *and* $h : A \to B$ *be the unique morphism in* $\mathbf{Alg}(\Sigma)$ *from the initial object;*
– $f : A \to P_\Delta(A)$ *be the unique arrow in* $\mathbf{Alg}(\Sigma)$ *from the initial object.*

Let us assume that for all equations $t_1 = t_2$ in E, with free variables $\{x_i\}_{i\in I}$, we have De Simone proofs as follows (for t_1, t_1', t_2, t_2' terms of T_Σ):

$$\frac{x_i \xrightarrow{l_i} y_i \quad i \in I}{t_1 \xrightarrow{l} t_1'} \quad \textit{implies} \quad \frac{x_i \xrightarrow{l_i} y_i \quad i \in I}{t_2 \xrightarrow{l} t_2'} \quad \textit{and} \quad t_1' =_E t_2' \tag{2}$$

and viceversa, only using the rules in R.

Then, g can be lifted to be a coalgebra on $\mathbf{Alg}(\Sigma)$*. It follows that bisimilarity in g is a congruence.*

3 A Labelled Transition System for Fusion Calculus

In this section, following the approach adopted in [2] for the pi-calculus, we provide a transition system lts_g for the fusion calculus and apply the general result recalled in Section 2.3 to lift lts_g to a bialgebra.

We first define a permutation algebra [2, 10] enriched with the operations of fusion calculus and with constants modelling explicit fusions $x = y$. Restriction ν corresponds to (x) in fusion calculus: it has no argument here, since the extruded or restricted name is assumed to be always the first one, i.e. x_0. Operators ρ are generic, finite name permutations, as described in Section 2.1; δ is meant to represent the substitution $[x_i \mapsto x_{i+1}]$, for $i = 0, 1, \ldots$. Of course, this substitution is not finite, but, at least in the case of an ordinary term p, it replaces a finite number of names, i.e., the free names of p.

The signature Σ also contains constants (regarded as operators with arity 0, of course) modelling recursive fusion terms and explicit fusions. The introduction of explicit fusions is intended to model substitutive effects of fusion calculus while keeping essentially the same permutation algebra as in [2]. Indeed, an explicit fusion $x = y$ allows to represent the global effect of a name fusion resulting from a synchronisation without need of replacing x to y or viceversa in the processes in parallel: names x and y can be used interchangeably in the context $x = y|\_$ In practice, rather than applying to an agent the substitutive effect of a fusion, the agent is run in parallel with the fusion itself. For this reason, the combination of explicit fusions and permutations is enough to accommodate (possibly, non-injective) substitutions within the permutation algebra model.

Definition 8. *A signature Σ for the fusion calculus is defined as follows:*

$$\Sigma ::= \mathbf{0} \mid \pi.\_ \mid \_|\_ \mid \nu.\_ \mid \rho\_ \mid \delta.\_ \mid x = y \mid c_{\mathtt{rec}X.P},$$

with prefixes $\pi ::= \bar{x}y, xy, \varphi$*, and for each fusion agent P.*

Definition 9 (permutation algebra for fusion calculus).
A permutation algebra B for fusion calculus is the initial algebra $B = T_{\Sigma,E}$ where:

- *Σ is the signature defined above;*
- *E is the set of axioms below:*

$$\begin{array}{ll}
\text{(par)} & p|\mathbf{0} \doteq p \quad p|q \doteq q|p \quad p|(q|r) \doteq (p|q)|r \\
\text{(res)} & \nu.\mathbf{0} \doteq \mathbf{0} \qquad \nu.(\delta.p)\,|\,q \doteq p|\nu.q \qquad \nu.\nu.[x_0 \leftrightarrow x_1]p \doteq \nu.\nu.p \\
\text{(group)} & (\rho' \circ \rho)p \doteq \rho'(\rho p) \qquad \mathrm{id}\, p \doteq p \\
\text{(perm)} & \rho\mathbf{0} \doteq \mathbf{0} \qquad \rho(\pi.p) \doteq \rho(\pi).\rho p \qquad \rho(p|q) \doteq \rho p\,|\,\rho q \\
 & \rho\nu.p \doteq \nu.\rho_{+1}p \qquad \rho\, c_{\mathtt{rec}X.P} \doteq c_{\rho(\mathtt{rec}X.P)} \\
\text{(delta)} & \delta.\mathbf{0} \doteq \mathbf{0} \qquad \delta.(\pi.p) \doteq \delta(\pi).\delta.p \qquad \delta.p\,|\,q \doteq (\delta.p)\,|\,\delta.q \\
 & \delta.\nu.p \doteq \nu.[x_0 \leftrightarrow x_1]\delta.p \qquad \delta.\, c_{\mathtt{rec}X.P} \doteq c_{\delta(\mathtt{rec}X.P)} \\
\text{(fus)} & x = x \doteq \mathbf{0} \quad \nu.(x_0 = x) \doteq \mathbf{0} \quad \rho(x = y) \doteq \rho(x) = \rho(y) \\
 & \delta.x = y \doteq \delta(x) = \delta(y)
\end{array}$$

In the above axioms, by $\rho(z)$ and $\delta(z)$, for $z = x, y$, we mean the syntactical application of permutations ρ and δ, respectively, to z; similarly, for $\rho(\pi)$ and $\delta(\pi)$. Axioms (**par**), and (**res**) correspond to the analogous axioms for fusion calculus. Axioms (**perm**) and (**delta**) rule how to invert the order of operators among each other, following the intuition that ν and δ decrease and increase variable indexes, respectively. By axioms (**fus**) permutations can be syntactically applied to explicit fusions and fusions of syntactically equal names are discarded. Notice that other expected properties like $\nu.\delta.p = p$ and $[x_0 \leftrightarrow x_1]\delta.\delta.p = \delta.\delta.p$ can be derived from these axioms.

We give below a translation of fusion agents into terms of algebra B. Then, we define a transition system lts_g for the algebra B and show that axioms in E satisfy the bisimilarity condition for lifting coalgebras to bialgebras, as required by Theorem 2. The translation is straightforward, except for restriction ν that gives the flavour of the De Brujin notation. The idea is to split standard restriction in three steps. First, one shifts all names up-wards to generate a fresh name x_0, then swaps $\delta(x)$ and x_0, and, finally, applies restriction on x_0, which now stands for what 'used to be' x.

Definition 10 (translation $[\![\cdot]\!]$). *We define a translation of fusion agents $[\![\cdot]\!] : \mathcal{F} \to |B|$ as follows:*

$$[\![\mathbf{0}]\!] = \mathbf{0} \qquad [\![\pi.P]\!] = \pi.[\![P]\!] \qquad [\![P|Q]\!] = [\![P]\!]\,|\,[\![Q]\!]$$
$$[\![(x)\,P]\!] = \nu.[\delta(x) \leftrightarrow x_0]\delta.[\![P]\!] \qquad [\![\mathtt{rec}X.P]\!] = c_{\mathtt{rec}X.P}$$

For example, the translation of a fusion agent $P = (x_2)(\{x_2 = x_4\}.\bar{x}_7 x_2.\mathbf{0})$ is $[\![P]\!] = \nu.\,[x_0 \leftrightarrow x_3]\delta.\,\{x_2 = x_4\}.\bar{x}_7 x_2.\mathbf{0} = \nu.\,[x_0 \leftrightarrow x_3]\{x_3 = x_5\}.\bar{x}_8 x_3.\mathbf{0} = \nu.\,\{x_0 = x_5\}.\bar{x}_8 x_0.\mathbf{0}$.

Definition 11. *We let L be the set of labels $L = \Lambda \times \Phi$, where $\Lambda = \{xy,\ \bar{x}y,\ \varphi,\ - \mid x, y, \mathrm{n}(\varphi) \in \mathfrak{N}\}$ and $-$ denotes a 'null' action, and Φ is the set of all fusions over $\mathfrak{N}$. We let $\alpha, \beta, \ldots$ range over Λ.*

By $\delta(\alpha)$ and $\nu(\alpha)$ we denote the labels obtained from α by respectively applying substitutions δ and ν to its names, where $\nu(x_{i+1}) = x_i$, $\delta(x_i) = x_{i+1}$, and in $\nu(\varphi)$ the equivalence class of x_0 is a singleton. The correspondence between the left components of the labels L and the actions of the fusion calculus is the obvious one for xy, $\bar{x}y$, φ. In the case of bound input transitions, only the channel x on which the output occurs is observed in label $\bar{x}$, and similarly in the case of bound input transitions x. On the other side, the right-hand compontent of each label in L are used to observe the fusions of names at that computational step and any small smaller fusion.

Definition 12. *The* entailment relation $\vdash$ *is defined as follows:* $\varphi \vdash \alpha = \beta$*, if* $\alpha, \beta \neq \psi$ *and* $\sigma(\alpha) = \sigma(\beta)$*, for a substitutive effect* σ *of* φ*;* $\varphi \vdash \psi = \psi'$ *if* $\varphi + \psi = \varphi + \psi'$.

Definition 13 (transition specification Δ). *The transition specification Δ is the tuple $\langle \Sigma, L, R \rangle$, where the signature Σ is as in Definition 9, labels L are defined in Definition 11 and R is the set of SOS rules in Table 2. Transitions take the form $p \xrightarrow{(\alpha,\varphi)} q$, where (α, φ) ranges over L.*

The first group of rules in Table 2 is essentially the same as that one given in [2] for the pi-calculus. The most interesting among them are in the right-column and concern bound I/O actions: they follow the intuition that substitutions on the source of a transition must be reflected on its destination by restoring the extruded or fresh name to x_0. Thus, for example, rule (Delta′) applies δ to q and then permutes x_0 and x_1, in order to have the extruded name back to x_0. Conversely, rule (Res′) permutes x_0 and x_1 to make sure that the restriction operation applies to x_0 and not to the extruded name x_1. In rule (Par′) side condition $\mathrm{bn}(\alpha) \cap \mathrm{fn}(r) = \emptyset$ is *not necessary*, since δ shifts any name in r to the right and, thus, x_0 does not appear in $\delta.r$.

The main novelty is the second group of rules in Table 2: such rules are suited to deal with explicit fusions. By rule (Exp) explicit fusions are propagated and by rules (Par$_1$), (Par$'_1$), and (Par$_f$) they are combined with each other and with other agents in parallel. Rules (Pre), (Pre′), and (Fus) are intended to ensure that the associated bisimilarity be preserved by closure with respect to fusions running in parallel. The presence of two rules (Pre) and (Pre′) is required in order to have a fully abstract translation of fusion agents into terms of the algebra, as shown in Example 3 below. All the side conditions are meant to ensure a saturation of process behaviours with respect to the explicit fusions. This form of saturation is formalised in the following proposition.

First we introduce the notion of equivalence relation $\mathtt{Eq}(p)$, induced by the explicit fusions in a term p. The notation given for fusions also applies to $\mathtt{Eq}(p)$: this holds in particular for $\nu(\mathtt{Eq}(p))$, $\delta(\mathtt{Eq}(p))$, and $\mathtt{Eq}(p) \vdash \alpha = \beta$.

Definition 14. *Let p be a term of algebra B. The equivalence relation $\mathtt{Eq}(p)$ obtained as the sum of all explicit fusions in p is inductively defined as follows:*

$$\mathtt{Eq}(\mathbf{0}) = \tau \qquad \mathtt{Eq}(\pi.p) = \tau \qquad \mathtt{Eq}(p|q) = \mathtt{Eq}(p) + Eq(q) \qquad \mathtt{Eq}(\nu.\,p) = \nu(\mathtt{Eq}(p))$$
$$\mathtt{Eq}(\rho p) = \rho(\mathtt{Eq}(p)) \quad \mathtt{Eq}(\delta.\,p) = \delta(\mathtt{Eq}(p)) \quad \mathtt{Eq}(x = y) = \{x = y\} \quad \mathtt{Eq}(c_{\mathtt{rec}X.P}) = \tau$$

For example, for $p \triangleq x = y\,|\,y = z\,|\,q$, $\mathtt{Eq}(p) = \{x = y = z\} + \mathtt{Eq}(q)$.

Proposition 3.

1. *If $p \xrightarrow{(\alpha,\varphi)} q$ then $p \xrightarrow{(\beta,\mathtt{Eq}(p)+\varphi)} q$, for all β such that $\mathtt{Eq}(p) + \varphi \vdash \alpha = \beta$.*
2. *If $p \xrightarrow{(\alpha,\varphi)} q$ then $p \xrightarrow{(\alpha,\psi)} q$, for all ψ such that $\psi \sqsubseteq \varphi$.*

Example 2.

1. The terms $p_1 \triangleq x = y\,|\,y = k\,|\,p$ and $p_2 \triangleq x = y\,|\,x = k\,|\,p$ have the same transitions. For instance, if $p_1 \xrightarrow{(\alpha,y=k)}$ then, by rules (Exp) and (Par$_f$), $p_2 \xrightarrow{(\alpha,\varphi)}$, for any $\varphi \sqsubseteq x = y + x = k$ and, in particular, for $\varphi = y = k$.
2. Consider $p = \bar{x}y.p_1\,|\,zk.p_2$. By rules (Pre) and (Com), $p \xrightarrow{(y=k,\varphi)} p_1\,|\,p_2\,|\,\psi'\,|\,y = k$, for all φ and ψ such that $x = z \sqsubseteq \psi$ and $\varphi \sqsubseteq \psi + y = k$; in other words, a synchronisation in p can take place in any context where x and z can be used interchangeably and, moreover, any 'smaller' fusion φ can be observed.

Proposition 4. *Let $\Delta = \langle \Sigma, L, R\rangle$ be the transition specification in Definition 13. Rules R are in De Simone format.*

Definition 15 (transition system lts_g). *The transition system for algebra B is $lts_g = \langle B, \longrightarrow\rangle$, where $\longrightarrow$ is defined by the SOS rules in Table 2 plus the following axiom:*

$$(\textsc{Rec})\ \frac{[\![P[\mathtt{rec}X.P/X]]\!] \xrightarrow{(\alpha,\varphi)} q}{c_{\mathtt{rec}X.P} \xrightarrow{(\alpha,\varphi)} q}$$

In fact, axiom (Rec) is an axiom schema, that is, for each constant $c_{\mathtt{rec}X.P}$, there is an axiom instance that provides a concrete way to build all the possible transitions that $c_{\mathtt{rec}X.P}$ undergoes; the fact that recursion is guarded ensures (Rec) to be well defined. Note that any axiom instance is in De Simone format. Moreover it can be proved that, for each fusion agent P, the number of constants and associated axioms (Rec) needed in all derivations of $[\![P[\mathtt{rec}X.P/X]]\!]$ is finite, up to name permutations. The proof is analogous to the proof given in [2] for the pi-calculus.

$$(\textsc{Rho})\quad \frac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha \neq x,\bar{x}}{\rho p \xrightarrow{(\rho(\alpha),\rho(\varphi))} \rho q} \qquad\qquad (\textsc{Rho}')\quad \frac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha = \bar{x},x}{\rho p \xrightarrow{(\rho_{+1}(\alpha),\rho_{+1}(\varphi))} \rho_{+1} q}$$

$$(\textsc{Del})\quad \frac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha \neq x,\bar{x}}{\delta.\,p \xrightarrow{(\delta(\alpha),\delta(\varphi))} \delta.\,q} \qquad\qquad (\textsc{Del}')\quad \frac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha = \bar{x},x}{\delta.\,p \xrightarrow{(\delta(\alpha),\delta(\varphi))} [x_0 \leftrightarrow x_1]\delta.\,q}$$

$$(\textsc{Par})\quad \frac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha \neq x,\bar{x}}{p|r \xrightarrow{(\alpha,\varphi)} q|r} \qquad\qquad (\textsc{Par}')\quad \frac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha = x,\bar{x}}{p|r \xrightarrow{(\alpha,\varphi)} q|\delta.\,r}$$

$$(\textsc{Res})\quad \frac{p \xrightarrow{(\delta(\alpha),\delta(\varphi))} q \quad \alpha = \bar{x}y,xy}{\nu.\,p \xrightarrow{(\alpha,\varphi)} \nu.\,q} \qquad (\textsc{Res}')\quad \frac{p \xrightarrow{(\delta(\alpha),\delta(\varphi))} q \quad \alpha = \bar{x},x}{\nu.\,p \xrightarrow{(\alpha,\varphi)} \nu.\,[x_0 \leftrightarrow x_1]q}$$

$$(\textsc{Open})\quad \frac{p \xrightarrow{(xx_0,\delta(\varphi))} q \quad x \neq x_0}{\nu.\,p \xrightarrow{(x,\delta(\varphi))} q} \qquad\qquad (\textsc{Scope})\quad \frac{p \xrightarrow{(\varphi,\delta(\psi))} q}{\nu.\,p \xrightarrow{(\nu(\varphi),\psi)} \nu.\,q}$$

$$(\textsc{Com})\quad \frac{p_1 \xrightarrow{(xy,\varphi)} q_1 \quad p_2 \xrightarrow{(\bar{x}z,\varphi)} q_2}{p_1|p_2 \xrightarrow{(y=z,\varphi)} q_1|q_2|y=z} \qquad (\textsc{Com}')\quad \frac{p_1 \xrightarrow{(xy,\varphi)} q_1 \quad p_2 \xrightarrow{(\delta(\bar{x}),\delta(\varphi))} q_2}{p_1|p_2 \xrightarrow{(\tau,\varphi)} q_1|\nu.\,(q_2|\delta(y) = x_0)}$$

$$(\textsc{Close})\quad \frac{p_1 \xrightarrow{(x,\varphi)} q_1 \quad p_2 \xrightarrow{(\bar{x},\varphi)} q_2}{p_1|p_2 \xrightarrow{(\tau,\nu(\varphi))} \nu.\,(q_1|q_2)}$$

$$(\textsc{Pre})\quad xy.p \xrightarrow{(x'y',\varphi')} p|\varphi \qquad \varphi' \sqsubseteq \varphi;\ \varphi \vdash xy = x'y'$$

$$(\textsc{Pre}')\quad xy.p \xrightarrow{(\delta(x'),\delta(\varphi'))} (\delta.\,(p|\varphi))\,|\,x_0 = \delta(y) \qquad \varphi' \sqsubseteq \varphi;\ \varphi \vdash x = x'$$

$$(\textsc{Fus})\quad \varphi.p \xrightarrow{(\varphi',\psi')} p|\psi + \varphi \qquad \psi' \sqsubseteq \psi;\ \psi \vdash \varphi = \varphi'$$

$$(\textsc{Exp})\quad x = y \xrightarrow{(-,x=y)} x = y \qquad x \neq y$$

$$(\textsc{Par}_1)\quad \frac{p_1 \xrightarrow{(\alpha,\varphi_1)} q_1 \quad p_2 \xrightarrow{(-,\varphi_2)} q_2 \quad \alpha = \bar{x}y,xy}{p_1|p_2 \xrightarrow{(\beta,\varphi')} q_1|q_2} \qquad \varphi' \sqsubseteq \varphi_1 + \varphi_2;\ \varphi_1 + \varphi_2 \vdash \alpha = \beta$$

$$(\textsc{Par}_1')\quad \frac{p_1 \xrightarrow{(\alpha,\varphi_1)} q_1 \quad p_2 \xrightarrow{(-,\varphi_2)} q_2 \quad \alpha = \bar{x},x}{p_1|p_2 \xrightarrow{(\beta,\varphi')} q_1|\delta.\,q_2} \qquad \varphi' \sqsubseteq \varphi_1 + \delta(\varphi_2);\ \varphi_1 + \delta(\varphi_2) \vdash \alpha = \beta$$

$$(\textsc{Par}_f)\quad \frac{p_1 \xrightarrow{(-,\varphi_1)} q_1 \quad p_2 \xrightarrow{(-,\varphi_2)} q_2}{p_1|p_2 \xrightarrow{(-,\varphi')} q_1|q_2} \qquad \varphi' \sqsubseteq \varphi_1 + \varphi_2$$

Rules (Pre), (Pre′), and (Open) are analogous with output actions; rule (Com′) has a symmetric counterpart.

Table 2. Structural Operational Semantics

We remark that $\sim_g$ is very close in spirit to the *inside-outside* bisimulation defined in [6], which satisfies the following properties: two equivalent terms must contain the same explicit fusions; two equivalent names must behave in the same way under explicit fusions context.

Theorem 3. *Let B be the permutation algebra defined in Definition 9. Then, Condition 2 in Theorem 2 holds.*

Proof (Hint). The proof consists in showing that for each $t_1 = t_2$ in E, for each $t_1 \xrightarrow{(\alpha,\varphi)} p$ there exists $t_2 \xrightarrow{(\alpha,\varphi)} q$ with $p \doteq q$, and viceversa. The proof is quite long, because there are several cases that have to be taken into account, depending on the transition rules that can be applied to t_1 and t_2, for all $t_1 = t_2$. In most cases the proof is analogous to the proof given in [2] for the pi-calculus. Here we only consider axiom $\nu.(\delta.p)|q = p|\nu.q$, that is one of the most interesting axioms and it also involves explicit fusions.

Suppose that, by rule (RES), $\nu.(\delta.p)|q \xrightarrow{(\alpha,\varphi)} \nu.p'$. Necessarily, $(\delta.p)|q \xrightarrow{(\delta(\alpha),\delta(\varphi))} p'$ and there is a number of possible cases. Suppose that by rule (PAR$_1$) $\delta.p \xrightarrow{(\delta(\beta),\delta(\varphi_1))} p''$, $q \xrightarrow{(-,\delta(\varphi_2))} q'$, with $\varphi \sqsubseteq \varphi_1 + \varphi_2$ and $\varphi_1 + \varphi_2 \vdash \alpha = \beta$; moreover $p' = p''|q'$. Then, $p \xrightarrow{(\beta,\varphi_1)} p'''$, with $p'' = \delta.p'''$. Now consider $p|\nu.q$. By rule (RES) $\nu.q \xrightarrow{(-,\varphi_2)} \nu.q_2$ and, by rule (PAR$_1$), $p|\nu.q \xrightarrow{(\alpha,\varphi)} p'''|\nu.q'$. And $\nu.p' \doteq p'''|\nu.q'$.

Corollary 2. *Let B be the algebra defined in Definition 9. Bisimilarity is a congruence in $g : B \to P_\Delta(B)$.*

Proof. It follows by Theorems 3 and 2.

Our next claim is that the translation $[\![\cdot]\!]$ of fusion agents into terms of the permutation algebra B is fully abstract with respect to hyperequivalence. Here we provide the reader with some intuition behind the proof. The formal proof is given in the appendix.

Theorem 4. *Let P and Q be two fusion agents. Then, $P \sim_{he} Q$ iff $[\![P]\!] \sim_g [\![Q]\!]$.*

Proof (Hint). The proof relies on the definition of three intermediate transition systems and their notions of bisimulation.

The first transition system lts_1 is defined by the rules given in Table 3. The rules are similar to those given in Table 2 for lts_g, except for the fact that lts_1 aims at ensuring saturation of behaviours of a term only with respect to the explicit fusion contained in the term, rather than with respect to all the possible fusion context. This difference shows up in rules for prefixes and communication. Moreover, lts_1 contains the rule

$$\text{(EQ)}\ \frac{p \xrightarrow{\alpha}_1 p' \quad \mathtt{Eq}(p) \vdash \alpha = \beta}{p \xrightarrow{\beta}_1 p'},$$

which replaces the rules for propagation and combination of explicit fusions ((EXP), (PAR$_1$), (PAR$'_1$), (PAR$_f$)). The fact that (EQ) has the same effect of the above rules easily follows by observing that by rule (EQ) lts_1 enjoys a special case of the saturation property

(Proposition 3.1) with $\varphi = \tau$. Thus, for example, $x = y \,|\, \bar{x}z.\mathbf{0}$ has a transition $\xrightarrow{\bar{x}z}_1$ as well as $\xrightarrow{\bar{y}z}_1$. The notion of bisimilarity $\sim_1$ is the standard one, except for the fact that bisimilar processes are also required to contain the same explicit fusions. Our first claim is that, for P and Q two fusion agents, $P \dot{\sim} Q$ if and only if $[\![P]\!] \sim_1 [\![Q]\!]$, being $\dot{\sim}$ the notion of fusion bisimulation given in Definition 1.

Our next step is to define a second transition system lts_2 by adding to lts_1 a rule for closing with respect to fusions in parallel:

$$\text{(CTX)}\ \frac{p \,|\, \varphi \xrightarrow{\alpha}_1 q}{p \xrightarrow{\alpha,\varphi}_2 q}$$

Bisimilarity $\sim_2$ is analogous to $\sim_1$ (with $\longrightarrow_2$ in place of $\longrightarrow_1$). We argue that, $P \sim_{he} Q$ if and only if $[\![P]\!] \sim_2 [\![Q]\!]$, where $\sim_{he}$ denotes fusion hyperequivalence. The intuition behind this result is that we are able to model in $\sim_2$ closure with respect to substitution, by considering any possible fusion context (rule (CTX)).

The third transition system lts_3 is defined by essentially the same rules as those given for lts_g, except for the fact that, akin to lts_1, a rule

$$\text{(EQ)}\ \frac{p \xrightarrow{(\alpha,\varphi)}_3 p' \quad \mathtt{Eq}(p) \vdash \alpha = \beta}{p \xrightarrow{(\beta,\varphi)}_3 p'},$$

replaces the rules for fusion propagation. Bisimilarity $\sim_3$ is analogous to $\sim_2$ ($\longrightarrow_3$ replaces $\longrightarrow_2$).

The proof of the theorem is concluded by showing that $\sim_3$ is equivalent to both $\sim_2$ and $\sim_g$. As to the equivalence of $\sim_2$ and $\sim_3$, the intuition is that $\sim_2$ and $\sim_3$ are both contained in $\sim_1$ and are preserved by fusion contexts: this is achieved in lts_2 by means of rule (CTX), while in lts_3 by the rules for prefixes.

Finally, the idea behind the proof of the equivalence of $\sim_3$ and $\sim_g$ is that lts_3 satisfies the saturation property stated in Proposition 3, by means of the combined use of rules for prefixes and (EQ).

Example 3. Consider two fusion agents $P = (x_2)\,(\{x_2 = x_4\}.\bar{x}_7x_2.\mathbf{0})$ and $Q = \tau.\bar{x}_7x_4.\mathbf{0}$. Of course, P and Q are hyperequivalent. Let us now translate P and Q in terms of algebra B. Then, $[\![P]\!] = \nu.\,\{x_0 = x_5\}.\bar{x}_8x_0.\mathbf{0}$ and $[\![Q]\!] = \tau.\bar{x}_7x_4.\mathbf{0}$. It holds that $[\![P]\!] \sim_g [\![Q]\!]$: the most interesting case is as follows. If $[\![P]\!] \xrightarrow{(\tau,\tau)} \nu.(\bar{x}_8x_0.\mathbf{0} \,|\, x_0 = x_8)$ then $[\![Q]\!] \xrightarrow{(\tau,\tau)} \bar{x}_7x_4.\mathbf{0}$. Next, if $\nu.(\bar{x}_8x_0. \,|\, x_0 = x_8) \xrightarrow{(\bar{x}_8,\tau)} x_0 = x_5$ then, by rule (PRE′), $\bar{x}_7x_4.\mathbf{0}$ is able to take the same step.

4 Conclusions

In this paper we have provided a compositional coalgebraic model of the Fusion calculus, following the approach applied in [2] to the pi-calculus. We have introduced a permutation algebra with the operations of the Fusion calculus along with constants modelling explicit fusions and recursion. We have then defined a labelled transition system for the enriched permutation algebra and we have proved that the conditions required by the result presented in [2] for lifting calculi with structural axioms to bialgebras are satisfied. It is worth noting that the introduction of explicit fusions allows to model name substitutions, while keeping essentially the same permutation algebra defined in [2] for the pi-calculus.

We are also studying a bialgebraic model of the open pi-calculus semantics [15]. We argue that the general theory developed in [2] cannot be straightforwardly applied to the open pi-calculus, because of the notion of 'distinction', that is needed in open pi-calculus to keep extruded names distinct from all (free) names. Our proposal is to extend the above theory with types, by defining an underlying multi-sorted permutation algebra, whose sorts are the distinctions.

A further challenge would be to consider general substitutions (on some first order signature), yielding models rather close to logic programming. We expect that the approach in [2] be flexible enough to allow varying the underlying algebra while employing similar constructions.

References

1. L. Aceto, W.J. Fokkink, and C. Verhoef. Structural Operational Semantics. In *Handbook of Process Algebra*, Elsevier, 2002.
2. M. Buscemi and U. Montanari. A First Order Coalgebraic Model of Pi-Calculus Early Observational Equivalence. In *Proc. of CONCUR '02*, LNCS 2421. Springer, 2002.
3. M. Buscemi and U. Montanari. A Compositional Coalgebraic Model of Monadic Fusion Calculus. TR-05-17, Dipartimento di Informatica, University of Pisa, 2005.
4. A. Corradini, M. Große-Rhode, R. Heckel. Structured transition systems as lax coalgebras. In *Proc. of CMCS'98*, ENTS 11. Elsevier Science, 1998.
5. A. Corradini, R. Heckel, and U. Montanari. Compositional SOS and beyond: A coalgebraic view of open systems. *Theoretical Computer Science* 280:163-192, 2002.
6. P. Gardner and L. Wischik. Explicit Fusions. In *Proc. of MFCS '00*, LNCS 1893. Springer-Verlag, 2002. Full version to appear in *Theoretical Computer Science*.
7. P. Gardner, C. Laneve, and L. Wischik. The fusion machine (extended abstract). In *Proc. of CONCUR '02*, LNCS 2421. Springer-Verlag, 2002.
8. L. G. Meredith, S. Bjorg, and D. Richter. Highwire Language Specification Version 1.0. Unpublished manuscript.
9. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (parts I and II). *Information and Computation*, 100(1):1–77, 1992.
10. U. Montanari and M. Pistore. Pi-Calculus, Structured Coalgebras and Minimal HD-Automata. In *Proc. of MFCS'00*, LNCS 1983. Springer, 2000.
11. U. Montanari and M. Pistore. Structured Coalgebras and Minimal HD-Automata for the pi-Calculus. Technical Report 0006-02, IRST-ITC, 2000. Available at the URL: `http://sra.itc.it/paper.epl?id=MP00`. Full, revised version to appear in *Theoretical Computer Science*.
12. J. Parrow and B. Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. In *Proc. of LICS'98*. IEEE Computer Society Press, 1998.
13. G. D. Plotkin. A Structural Approach to Operational Semantics. *Journal of Logic and Algebraic Programming*, 60-61: 17–139, 2004.
14. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science* 249(1):3–80, 2000.

15. D. Sangiorgi. A Theory of Bisimulation for the pi-Calculus. *Acta Informatica*, 33(1): 69-97, 1996.
16. D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. of LICS'97*, IEEE. Computer Society Press, 1997.
17. B. Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. PhD thesis, Department of Computer Systems, Uppsala University, 1998.
18. L. Wischik and P. Gardner. Strong Bisimulation for the Explicit Fusion Calculus. In *Proc. of FoSSaCS '04*, LNCS 2987. Springer, 2004.

A Proof of Theorem 4

We now give a formal proof of Theorem 4, by detailing the steps that we have outlined in Section 3.

Definition 16. *Let σ be a substitution. The syntactical application of σ to any term p of algebra B is inductively defined as follows:*

$$\sigma(\mathbf{0}) = \mathbf{0} \qquad \sigma(\pi.p) = \sigma(\pi).\sigma(p) \qquad \sigma(p|q) = \sigma p|\sigma q \qquad \sigma(\nu.\, p) = \nu.\, \sigma_{+1} p$$
$$\sigma(\rho p) = (\sigma \circ \rho)(p) \qquad \sigma(\delta.\, p) = \sigma \circ \delta.\, (p) \qquad \sigma(x = y) = \sigma(x) = \sigma(y)$$
$$\sigma(c_{\mathtt{rec}X.P}) = c_{\sigma(\mathtt{rec}X.P)}$$

Note that $\sigma(p)$ contains no explicit fusion if σ is a substitutive effect of $\mathtt{Eq}(p)$.

Definition 17. *The transition system lts_1 is defined as $lts_1 = \langle B, \longrightarrow_1 \rangle$, where $\longrightarrow_1$ is defined by the rules in Table 3.*

As mentioned, the rules in Table 3 are similar to those given in Table 2 for lts_g. There are two main differences. First, lts_1 aims at ensuring saturation of behaviours of a term only with respect to the explicit fusion contained in the term. Thus, rules for prefixes do not consider all the possible fusion contexts. Second, lts_1 contains rule (EQ) in place of (EXP), (PAR$_1$), (PAR$'_1$), and (PAR$_f$). In fact, it can be easily seen that rule (EQ) has the same effect of the above rules for propagation and combination of explicit fusions, but it is not in De Simone format.

Definition 18 (bisimilarity $\sim_1$). *A* bisimulation on lts_1 *is a binary symmetric relation $\mathcal{S}$ between terms of B such that $p\ \mathcal{S}\ q$ implies:*

1. $\mathtt{Eq}(p) = \mathtt{Eq}(q)$*;*
2. *for each $p \xrightarrow{\alpha}_1 p'$ there is some $q \xrightarrow{\beta}_1 q'$ such that $\mathtt{Eq}(p) \vdash \alpha = \beta$ and $p'\ \mathcal{R}\ q'$, and viceversa.*

Bisimilarity $\sim_1$ is the largest bisimulation on lts_1.

Our first claim is that, for P and Q two fusion agents, $P \dot{\sim} Q$ if and only if $[\![P]\!] \sim_1 [\![Q]\!]$, being $\dot{\sim}$ the notion of fusion bisimulation.

Proposition 5. *Let p and q be two terms of algebra B and let σ be a substitutive effect of $\mathtt{Eq}(p)$. Then, $p \sim_1 q$ if and only if $\sigma(p) \sim_1 \sigma(q)$.*

Proof. First, note that, for σ a substitutive effect of $\mathtt{Eq}(p)$, it holds that $\sigma(p) \xrightarrow{\alpha}_1 q$, with $\alpha = \sigma(\beta)$ and $q = \sigma(q')$, for some β and q'. Moreover, $\sigma(p) \xrightarrow{\sigma(\alpha)}_1$ iff $\rho(p) \xrightarrow{\rho(\alpha)}_1$, for any substitutive effects ρ of $\mathtt{Eq}(p)$.

Next, let p and q be two terms of algebra B and let σ be a substitutive effect of $\mathtt{Eq}(p)$. Then, $p \xrightarrow{\alpha}_1 q$ with $\alpha \neq \bar{x}, x$ (resp. $\alpha = \bar{x}, x$) if and only if $\sigma(p) \xrightarrow{\sigma(\alpha)}_1 \sigma(q)$ (resp. $\sigma(p) \xrightarrow{\sigma_{+1}(\alpha_{+1})}_1 \sigma_{+1}(q)$) and viceversa. This is easily proved by induction on the rules of lts_1. For instance, consider $p_1 \triangleq x = y \mid \bar{z}v$ and $p_2 \triangleq z = y \mid xw$. Since $p_1|p_2 \vdash z = x$, by rule (COM) $p|q \xrightarrow{\tau}_1$. On the other hand, any substitutive effect σ of $\mathtt{Eq}(p_1|p_2)$ fuses z and x and, thus, $\sigma(p_1|p_2) \xrightarrow{\tau}_1$. The thesis follows by the above remarks.

$$(\textsc{Rho})\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha \neq x,\bar{x}}{\rho p \xrightarrow{\rho(\alpha)}_1 \rho q} \qquad\qquad (\textsc{Rho}')\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha = \bar{x},x}{\rho p \xrightarrow{\rho_{+1}(\alpha)}_1 \rho_{+1} q}$$

$$(\textsc{Del})\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha \neq x,\bar{x}}{\delta.\,p \xrightarrow{\delta(\alpha)}_1 \delta.\,q} \qquad\qquad (\textsc{Del}')\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha = \bar{x},x}{\delta.\,p \xrightarrow{\delta(\alpha)}_1 [x_0 \leftrightarrow x_1]\delta.\,q}$$

$$(\textsc{Par})\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha \neq x,\bar{x}}{p|r \xrightarrow{\alpha}_1 q|r} \qquad\qquad (\textsc{Par}')\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha = x,\bar{x}}{p|r \xrightarrow{\alpha}_1 q|\delta.\,r}$$

$$(\textsc{Res})\quad \frac{p \xrightarrow{\alpha}_1 q \quad \alpha = \bar{x}y, xy;\ x_0 \neq x,y}{\nu.\,p \xrightarrow{\nu(\alpha)}_1 \nu.\,q} \qquad (\textsc{Res}')\quad \frac{p \xrightarrow{(\alpha,\delta(\varphi))} q \quad \alpha = \bar{x},x;\ x \neq x_0}{\nu.\,p \xrightarrow{\nu(\alpha)}_1 \nu.\,[x_0 \leftrightarrow x_1]q}$$

$$(\textsc{Open})\quad \frac{p \xrightarrow{xx_0}_1 q \quad x \neq x_0}{\nu.\,p \xrightarrow{x}_1 q} \qquad\qquad (\textsc{Scope})\quad \frac{p \xrightarrow{\varphi}_1 q}{\nu.\,p \xrightarrow{\nu(\varphi)}_1 \nu.\,q}$$

$$(\textsc{Rec})\quad \frac{[\![P[\mathtt{rec}X.P/X]]\!] \xrightarrow{\alpha}_1 q}{c_{\mathtt{rec}X.P} \xrightarrow{\alpha}_1 q}$$

$$(\textsc{Com})\quad \frac{p_1 \xrightarrow{xy}_1 q_1 \quad p_2 \xrightarrow{\bar{x}'z}_1 q_2}{p_1|p_2 \xrightarrow{y=z}_1 q_1|q_2|y=z} \quad p_1|p_2 \vdash x = x'$$

$$(\textsc{Com}')\quad \frac{p_1 \xrightarrow{xy}_1 q_1 \quad p_2 \xrightarrow{\delta(\bar{x}')}_1 q_2}{p_1|p_2 \xrightarrow{\tau}_1 q_1|\nu.\,(q_2|\delta(y) = x_0)} \quad p_1|p_2 \vdash x = x'$$

$$(\textsc{Close})\quad \frac{p_1 \xrightarrow{x}_1 q_1 \quad p_2 \xrightarrow{\bar{x}'}_1 q_2}{p_1|p_2 \xrightarrow{\tau}_1 \nu.\,(q_1|q_2)} \quad p_1|p_2 \vdash x = x'$$

$$(\textsc{Pre})\quad xy.p \xrightarrow{xy}_1 p \qquad\qquad (\textsc{Pre}')\quad xy.p \xrightarrow{\delta(x)}_1 \delta.\,p|\delta(y) = x_0$$

$$(\textsc{Fus})\quad \varphi.p \xrightarrow{\varphi}_1 p|\varphi \qquad\qquad (\textsc{Eq})\quad \frac{p \xrightarrow{\alpha}_1 q \quad \mathtt{Eq}(p) \vdash \alpha = \beta}{p \xrightarrow{\beta}_1 q}$$

(Open) is analogous with output actions; rule (Com′) has a symmetric counterpart.

Table 3. Transition System lts_1

Theorem 5. *Let p and q be two terms of algebra B and let σ be a substitutive effect of* $\mathtt{Eq}(p)$*. Then, $p \sim_1 q$ if and only if $\{[\sigma(p)]\} \dot{\sim} \{[\sigma(q)]\}$, where $\{[\cdot]\}$ is the inverse translation of $[\![\cdot]\!]$.*

Corollary 3. *Let P and Q be two fusion agents. $P \dot{\sim} Q$ if and only if $[\![P]\!] \sim_1 [\![Q]\!]$.*

Definition 19. *The transition system lts_2 is defined as $lts_2 = \langle B, \longrightarrow_2 \rangle$, where $\longrightarrow_2$ is defined by adding to the rules of lts_1 a rule for closing with respect to fusions in parallel:*

$$\text{(CTX)}\ \frac{p|\varphi \xrightarrow{\alpha}_1 q}{p \xrightarrow{\alpha,\varphi}_2 q}$$

Bisimulation and bisimilarity $\sim_2$ are analogous to those defined for lts_1, with $\longrightarrow_2$ in place of $\longrightarrow_1$.

Lemma 1. *Let p and q be two terms of algebra B. If $p \sim_2 q$ then $p \,|\, \varphi \sim_2 q \,|\, \varphi$, for all φ.*

Theorem 6. *$P \sim_{he} Q$ if and only if $[\![P]\!] \sim_2 [\![Q]\!]$, where $\sim_{he}$ denotes fusion hyperequivalence.*

Proof. The proof follows by Corollary 3 and by Lemma 1.

Definition 20. *The third transition system lts_3 is defined as $lts_3 = \langle B, \longrightarrow_3 \rangle$, where $\longrightarrow_3$ is given by the rules in Table 4. Bisimilarity $\sim_3$ is analogous to $\sim_2$ ($\longrightarrow_2$ replaces $\longrightarrow_3$).*

The proof of the Theorem 4 is concluded by showing that $\sim_3$ is equivalent to both $\sim_2$ and $\sim_g$.

Lemma 2. *Let p and q be two terms of algebra B. If $p \sim_3 q$ then $p \,|\, \varphi \sim_3 q \,|\, \varphi$, for all φ.*

Theorem 7. *Let p and q be two terms of algebra B. Then, $p \sim_3 q$ if and only if $p \sim_2 q$.*

Proof. It follows by the fact that $\sim_2 \subseteq \sim_1$ and $\sim_3 \subseteq \sim_1$ and by Lemmata 1 and 2.

Theorem 8. *Let p and q be two terms of algebra B. $p \sim_3 q$ if and only if $p \sim_g q$.*

Proof. The proof relies on the fact that the rules in lts_g for propagation and combination of explicit fusions ((EXP), (PAR$_1$), (PAR$'_1$), (PAR$_f$)) can be simulated rule (EQ) in lts_3 and viceversa. Moreover, lts_g ensures that if $p \sim_g q$ then $\mathtt{Eq}(p) = \mathtt{Eq}(q)$.

$$(\textsc{Rho})\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \alpha \neq x, \bar{x}}{\rho p \xrightarrow{(\rho(\alpha),\rho(\varphi))}_3 \rho q} \qquad (\textsc{Rho}')\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \alpha = \bar{x}, x}{\rho p \xrightarrow{(\rho_{+1}(\alpha),\rho_{+1}(\varphi))}_3 \rho_{+1} q}$$

$$(\textsc{Del})\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \alpha \neq x, \bar{x}}{\delta.\, p \xrightarrow{(\delta(\alpha),\delta(\varphi))}_3 \delta.\, q} \qquad (\textsc{Del}')\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \alpha = \bar{x}, x}{\delta.\, p \xrightarrow{(\delta(\alpha),\delta(\varphi))}_3 [x_0 \leftrightarrow x_1]\delta.\, q}$$

$$(\textsc{Par})\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \alpha \neq x, \bar{x}}{p|r \xrightarrow{(\alpha,\varphi)}_3 q|r} \qquad (\textsc{Par}')\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \alpha = x, \bar{x}}{p|r \xrightarrow{(\alpha,\varphi)}_3 q|\delta.\, r}$$

$$(\textsc{Res})\quad \frac{p \xrightarrow{(\delta(\alpha),\delta(\varphi))}_3 q \quad \alpha = \bar{x}y, xy}{\nu.\, p \xrightarrow{(\alpha,\varphi)}_3 \nu.\, q} \qquad (\textsc{Res}')\quad \frac{p \xrightarrow{(\delta(\alpha),\delta(\varphi))}_3 q \quad \alpha = \bar{x}, x}{\nu.\, p \xrightarrow{(\alpha,\varphi)}_3 \nu.\, [x_0 \leftrightarrow x_1] q}$$

$$(\textsc{Open})\quad \frac{p \xrightarrow{(xx_0,\delta(\varphi))}_3 q \quad x \neq x_0}{\nu.\, p \xrightarrow{(x,\delta(\varphi))}_3 q} \qquad (\textsc{Scope})\quad \frac{p \xrightarrow{(\varphi,\delta(\psi))}_3 q}{\nu.\, p \xrightarrow{(\nu(\varphi),\psi)}_3 \nu.\, q}$$

$$(\textsc{Com})\quad \frac{p_1 \xrightarrow{(xy,\varphi)}_3 q_1 \quad p_2 \xrightarrow{(\bar{x}z,\varphi)}_3 q_2}{p_1|p_2 \xrightarrow{(y=z,\varphi)}_3 q_1|q_2|y=z} \qquad (\textsc{Com}')\quad \frac{p_1 \xrightarrow{(xy,\varphi)}_3 q_1 \quad p_2 \xrightarrow{(\delta(\bar{x}),\delta(\varphi))}_3 q_2}{p_1|p_2 \xrightarrow{(\tau,\varphi)}_3 q_1|\nu.\, q_2|\delta(y) = x_0}$$

$$(\textsc{Close})\quad \frac{p_1 \xrightarrow{(x,\varphi)}_3 q_1 \quad p_2 \xrightarrow{(\bar{x},\varphi)}_3 q_2}{p_1|p_2 \xrightarrow{(\tau,\nu(\varphi))}_3 \nu.\, q_1|q_2} \qquad (\textsc{Rec})\quad \frac{[\![P[\mathtt{rec}\, X.\, P/X]]\!] \xrightarrow{(\alpha,\varphi)}_3 q}{c_{\mathtt{rec}\, X.\, P} \xrightarrow{(\alpha,\varphi)}_3 q}$$

$$(\textsc{Pre})\quad xy.p \xrightarrow{(x'y',\varphi')}_3 p|\varphi \qquad \varphi' \sqsubseteq \varphi;\ \varphi \vdash xy = x'y'$$

$$(\textsc{Pre}')\quad xy.p \xrightarrow{(\delta(x'),\delta(\varphi'))}_3 (\delta.\,(p|\varphi))\,|x_0 = \delta(y) \qquad \varphi' \sqsubseteq \varphi;\ \varphi \vdash x = x'$$

$$(\textsc{Fus})\quad \varphi.p \xrightarrow{(\varphi',\psi')}_3 p|\psi + \varphi \qquad \psi' \sqsubseteq \psi;\ \psi \vdash \varphi = \varphi'$$

$$(\textsc{Eq})\quad \frac{p \xrightarrow{(\alpha,\varphi)}_3 q \quad \mathtt{Eq}(p) \vdash \alpha = \beta}{p \xrightarrow{(\beta,\varphi)}_3 q}$$

Rules (Pre), (Pre′), and (Open) have analogous counterparts for output actions; rule (Com′) has a symmetric counterpart.

Table 4. Transition System lts_3