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 authors and previously applied to 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. Interestingly 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 [9, 12] has been introduced as a variant of the pi-calculus. It makes input and output operations fully symmetric and enables a more general name matching mechanism during synchronisation. The version with explicit fusions [4, 5] aims at propagating fusions to the environment in an asynchronous way. 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 [5], or forms of pattern matching between pairs of messages [6].

In fusion calculus [9, 12], a fusion, as soon as it is generated, it is immediately applied to the whole system and has the effect of a (possibly non-injective) name substitution. Explicit fusions [4], instead, are processes that exist concurrently with the rest of the system and enable to freely use two names one for the other. Interestingly enough, the combination of explicit fusions and restriction allows to derive a name substitution operator which behaves like the standard capture-avoiding substitution.

Interactive systems, when represented as labelled transition systems, can be conveniently modelled as coalgebras. A coalgebraic framework [10] 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 bisimulation associated to 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 [11, 2, 3] 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 [7, 8].

When considering more complex operations, the interaction of structural axioms with inference rules makes the application of the bialgebraic approach problematic. To overcome this difficulty, in [1] 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 result to the fusion calculus of Parrow and Victor, in order to provide a bialgebraic model of the calculus. 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 [1] are satisfied. Remarkably enough, explicit fusions enable us to model substitutions within our theory, while keeping essentially the same permutation algebra considered in [1] 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 hyperbisimulation. As in [13], closure with respect to substitution is obtained by adding in parallel at each step any possible fusion.

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 $\{x : x\,\varphi\,y \text{ for some } y \neq x\}$;
– ϵ denote the identity fusion (i.e., $\mathrm{n}(\epsilon) = \emptyset$);
– $\varphi+\psi$ denote the finest fusion which is coarser than φ and ψ, that is $(\varphi\cup\psi)^\star$;
– $\varphi \vdash \psi = \psi'$ denote that $\varphi + \psi = \varphi + \psi'$;
– 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 than 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. We abbreviate by $[y \leftrightarrow x]$ the substitution $[y \mapsto x, x \mapsto y]$. 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_φ agreeing with φ such that $\forall\, x, y : \sigma(x) = \sigma(y) \Rightarrow x\,\varphi\,y$ (i.e., σ sends all members of the equivalence class to one representative of the class). The only substitutive effect of a communication action is id.

A *name permutation* is a bijective name substitution. We use ρ to denote a permutation. 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 [9]. 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:

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

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.

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

(fus)	$\varphi.P \equiv \varphi.\sigma_\varphi(P)$ for σ_φ a substitutive effect of φ
(sum)	$P + \mathbf{0} \equiv P \qquad P + Q \equiv Q + P \qquad P + (Q + R) \equiv (P + Q) + R$
(par)	$P \vert \mathbf{0} \equiv P \qquad P \vert Q \equiv Q \vert P \qquad P \vert (Q \vert R) \equiv (P \vert Q) \vert R$
(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)	$P \vert (z)\,Q \equiv (z)\,(P \vert Q)$ 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 notation $\varphi \setminus z$ stands for the equivalence relation φ with all references to z removed (except for the identity).

The family of transitions $P \overset{\gamma}{\longmapsto} Q$ is the least family satisfying the laws in Table 1.

(F-Pre)	$\pi.P \overset{\pi}{\longmapsto} P$	(F-Sum)	$\dfrac{P \overset{\gamma}{\longmapsto} P'}{P{+}Q \overset{\gamma}{\longmapsto} P'}$ and symmetric
(F-Par)	$\dfrac{P \overset{\gamma}{\longmapsto} Q}{P \vert R \overset{\gamma}{\longmapsto} Q \vert R}$ if $\mathrm{bn}(\gamma) \cap \mathrm{fn}(R) = \emptyset$	(F-Com)	$\dfrac{P \overset{\bar{x}y}{\longmapsto} P' \quad Q \overset{xz}{\longmapsto} Q'}{P \vert Q \overset{\{y=z\}}{\longmapsto} P' \vert Q'}$
(F-Scope)	$\dfrac{P \overset{\varphi}{\longmapsto} Q \quad z\,\varphi\,x, z \neq x}{(z)\,P \overset{\varphi - z}{\longmapsto} [x/z]Q}$	(F-Open)	$\dfrac{P \overset{a\,z}{\longmapsto} Q \quad a \notin \{z, \bar{z}\}}{(z)\,P \overset{a(z)}{\longmapsto} Q}$
(F-Pass)	$\dfrac{P \overset{\gamma}{\longmapsto} P'}{(z)\,P \overset{\gamma}{\longmapsto} (z)\,P'}$ $z \notin \mathrm{n}(\gamma)$	(F-Rec)	$\dfrac{P[\mathtt{rec}\,X.\,P/X] \overset{\gamma}{\longmapsto} Q}{\mathtt{rec}\,X.\,P \overset{\gamma}{\longmapsto} Q}$
(F-Cong)	$\dfrac{P \equiv P' \quad P' \overset{\gamma}{\longmapsto} Q' \quad Q' \equiv Q}{P \overset{\gamma}{\longmapsto} Q}$		

Table 1. LTS for Fusion

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 $P \overset{\gamma}{\longmapsto} P'$ with $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q) = \emptyset$ then $Q \overset{\gamma}{\longmapsto} Q'$ and $\sigma_\gamma P' \ \mathcal{S} \ \sigma_\gamma Q'$ for some substitutive effect σ_γ of γ.*

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

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

We recall that an algebra A over a signature Σ (Σ-algebra in brief) is defined by a carrier set $|A|$ and, for each operation $op \in \Sigma$ of arity n, by a function $op^A : |A|^n \to |A|$. A 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*. The following definition introduces labelled transition systems whose states have an algebraic structure.

Definition 3 (transition systems). *Let Σ be a signature, and L be a set of labels. A* transition system *over Σ and L is a pair $lts = \langle A, \longmapsto_{lts} \rangle$ where A is a nonempty Σ-algebra and $\longmapsto_{lts} \subseteq |A| \times L \times |A|$ is a labelled transition relation. For $\langle p, l, q\rangle \in \longmapsto_{lts}$ we write $p \stackrel{l}{\longmapsto}_{lts} q$.*

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

The notion of bisimulation on structured transition systems is the classical one.

Definition 4 (bisimulation). *Let Σ be a signature, L be a set of labels, and $lts = \langle A, \longmapsto_{lts} \rangle$ be a transition system over Σ and L.*

A relation $\mathcal{R}$ over $|A|$ is a bisimulation *if $p \mathrel{\mathcal{R}} q$ implies:*

- *for each $p \stackrel{l}{\longmapsto} p'$ there is some $q \stackrel{l}{\longmapsto} q'$ such that $p' \mathrel{\mathcal{R}} q'$;*
- *for each $q \stackrel{l}{\longmapsto} q'$ there is some $p \stackrel{l}{\longmapsto} p'$ such that $p' \mathrel{\mathcal{R}} q'$.*

Bisimilarity *$\sim_{lts}$ is the largest bisimulation.*

Given a signature Σ and a set of labels L, a collection of SOS rules can be regarded as a specification of those transition systems over Σ and L that have a transition relation closed under the given rules.

Definition 5 (SOS rules). *Given a signature Σ and a set of labels L, a sequent $p \stackrel{l}{\longmapsto} q$ (over L and Σ) 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 Σ and L takes the form:*

$$\frac{p_1 \stackrel{l_1}{\longmapsto} q_1 \quad \cdots \quad p_n \stackrel{l_n}{\longmapsto} q_n}{p \stackrel{l}{\longmapsto} q}$$

where $p_i \stackrel{l_i}{\longmapsto} q_i$ as well as $p \stackrel{l}{\longmapsto} q$ are sequents.

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

---

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

We represent with

$$\frac{\dfrac{p_1 \xmapsto{l_1} q_1 \quad \cdots \quad p_n \xmapsto{l_n} q_n}{\vdots}}{p \xmapsto{l} q}$$

a proof*, with premises* $p_i \xmapsto{l_i} q_i$ *for* $i = 1, \ldots, n$ *and conclusion* $p \xmapsto{l} q$*, obtained by applying the rules in* R*.*

Definition 6 (transition specifications). *A* transition specification *is a tuple* $\Delta = \langle \Sigma, L, R \rangle$ *consisting of a signature* Σ*, a set of labels* L*, and a set of SOS rules* R *over* Σ *and* L*.*

A transition system over Δ *is a transition system over* Σ *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 [10].

Definition 7 (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* $h\ ;\ g = f\ ;\ F(h)$*. 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* $f_{lts}(p) = \{\langle l, q \rangle \mid p \xmapsto{l}_{lts} q\} \cup \{p\}$ *and, conversely, by* $p \xmapsto{l}_{lts_f} q$ *if and only if* $\langle l, q \rangle \in f(p)$*.*

Definition 8 (De Simone format). *Given a signature* Σ *and a set of labels* L*, a rule* r *over* Σ *and* L *is in* De Simone format *if it has the form:*

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

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* $i \notin I$*.*

The following results are due to [11] and concern *bialgebras*, i.e., coalgebras in $\mathbf{Alg}(\Sigma)$. 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, as noted in the introduction.

[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.

Proposition 2 (lifting of P_L). *Let $\Delta = \langle \Sigma, L, R\rangle$ be a transition specification with rules in De Simone format.*

Define $P_\Delta : \mathbf{Alg}(\Sigma) \to \mathbf{Alg}(\Sigma)$ as follows:

- $|P_\Delta(A)| = P_L(|A|)$;
- *whenever* $\dfrac{\{x_i \overset{l_i}{\longmapsto} y_i \mid i \in I\}}{op(x_1, \ldots, x_n) \overset{l}{\longmapsto} p} \in R$ *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, in order to prove that bisimilarity is a congruence, Corollary 1 requires that the lifting of a P_L-coalgebra to be P_Δ-coalgebra takes place. In fact, this step 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 results are due to [1] and generalise the theory described so far to algebras with structural axioms. In particular, Theorem 2 below states that a lts with structural axioms can be lifted from $\mathbf{Coalg}(P_L)$ to $\mathbf{Coalg}(P_\Delta)$ under appropriate conditions. Hence, it follows by Corollary 1 that bisimilarity is a congruence in this more general settings.

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* $B = T_{(\Sigma \cup C, E)}$.
2. *lts_g satisfies transition specification $\Delta = (\Sigma, L, R)$, with R in De Simone format.*
3. *A set of basic transitions $T \subseteq C \times L \times T_{\Sigma \cup C}$ exists for constants C, namely, $(c, l, t) \in T$ implies $c \xrightarrow{l}_g [t]_E$.*

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 rules:

$$\text{(Const)}\ \frac{(c,l,t) \in T}{c \xrightarrow{l}_{\hat{g}} t} \qquad \text{(Struct)}\ \frac{t_1 =_E t'_1 \quad t'_1 \xrightarrow{l}_{\hat{g}} t'_2 \quad t'_2 =_E t_2}{t_1 \xrightarrow{l}_{\hat{g}} t_2}$$

where terms t, t_1, t'_1, t_2, t'_2 are in $T_{\Sigma \cup C}$.

Definition 9. *Let $g : |B| \to P_L(|B|)$ be the initial coalgebra of Theorem 1, where, however, constants C are considered as auxiliary, i.e., B is seen as a Σ-algebra. Then, we define the following Σ-algebras and Σ-morphisms:*

- *$A = T_\Sigma(C)$ and $h : A \to B$ as the unique extension in $\mathbf{Alg}(\Sigma)$ of $h(c) = [c]_E$, for $c \in C$;*
- *$f : A \to P_\Delta(A)$ as the unique extension of $f(c) = \{(l,t) \mid (c,l,t) \in T\}$ in $\mathbf{Alg}(\Sigma)$.*

Theorem 2. *Let g be the initial coalgebra in $\mathcal{B}$ as specified by Theorem 1, and let A, h, and f be defined as in Definition 9. Then, h is surjective. 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:*

$$\frac{x_i \xmapsto{l_i} y_i \quad i \in I}{t_1 \xmapsto{l} t'_1} \quad \textit{implies} \quad \frac{x_i \xmapsto{l_i} y_i \quad i \in I}{t_2 \xmapsto{l} t'_2} \quad \textit{and} \quad t'_1 =_E t'_2 \tag{1}$$

and viceversa, using the rules in R and the additional rules:

$$c \xmapsto{l} t \quad \textit{iff} \quad (c,l,t) \in T.$$

Then, the left diagram below commutes in $\mathbf{Set}$, i.e., $h;g = f;P_L(h)$. Thus, the right diagram commutes in $\mathbf{Alg}(\Sigma)$ and g can be lifted from $\mathbf{Set}$ to $\mathbf{Alg}(\Sigma)$.

$$\begin{tikzcd} {|A|} \arrow[r, "h"] \arrow[d, "f"'] & {|B|} \arrow[d, "g"] \\ P_L(|A|) \arrow[r, "P_L(h)"'] & P_L(|B|) \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] & B \arrow[d, "g"] \\ P_\Delta(A) \arrow[r, "P_\Delta(h)"'] & P_\Delta(B) \end{tikzcd}$$

Corollary 2. *Let g be the initial coalgebra in $\mathcal{B}$ as specified by Theorem 1, and suppose g can be lifted from $\mathbf{Set}$ to $\mathbf{Alg}(\Sigma)$. Then in g bisimilarity is a congruence.*

3 A Labelled Transition Systems for Fusion Calculus

In this section, following the approach adopted in [1] for the pi-calculus, we provide a structured labelled transition system lts_g for the fusion calculus and apply the general result recalled in Subsection 2.3 to lift lts_g to be a bialgebra. It follows that bisimilarity in lts_g is a congruence.

We first define a permutation algebra enriched with the operations of fusion calculus and with explicit fusions $x = y$. Operators ρ are generic, finite name permutations, as described in Subsection 2.1; δ is meant to represent the substitution $[x_i \mapsto x_{i+1}]$, for $i = 0, 1, \ldots$. Restriction ν (corresponding to (x) in fusion calculus) has no argument, since the extruded or restricted name is assumed to be always the first one, i.e. x_0. By $\rho(x)$ and $\delta\,(x)$ we denote the syntactical application respectively of ρ and δ to name x; homomorphically, $\rho(\cdot)$ and $\delta\,(\cdot)$ are extended to fusions.

The introduction of explicit fusions in the signature Σ is intended to model substitutive effects of fusion calculus while keeping essentially the same permutation algebra as in [1]. In fact, 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 $x = y|\_$.

Definition 10 (permutation algebra for fusion calculus). *A permutation algebra B for fusion calculus is the initial algebra $B = T_{\Sigma \cup C, E}$ where:*

- *signature Σ is defined as follows:*

$$\Sigma \ ::= \ \mathbf{0} \ \big| \ \pi.\_ \ \big| \ \_+\_ \ \big| \ \_|\_ \ \big| \ \nu.\_ \ \big| \ \rho\_ \ \big| \ \delta.\_ \ \big| \ x = y,$$

  with prefixes $\pi \ ::= \ \bar{x}y, xy, \varphi$;
- *C is the set of constants*

$$C = \{c_{\mathtt{rec}\,X.\,P} \mid P \textit{ is a fusion agent }\};$$

- *E is the set of axioms below:*

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

Axioms (**sum**), (**par**), and (**res**) correspond to the analogous axioms for fusion calculus. The other axioms rule how to invert the order of operators among each other, following the intuition that ν and δ decrease and increase variable indexes, respectively. 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. 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 upwards 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 11 (translation $[\![\cdot]\!]$). *We define a translation of fusion agents $[\![\cdot]\!] : F \to |B|$ as follows:*

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

We now define a transition system lts_g for the above algebra and show that it satisfies the conditions required by Theorem 2.

Definition 12. *Let Λ be the set $\Lambda = \{x\,y,\ x,\ \overline{x}\,y,\ \overline{x},\ \varphi \mid x, y, \mathrm{n}(\varphi) \in \mathfrak{N}\}$ and Φ be the set of all fusions over $\mathfrak{N}$. We define the set L of labels as $L = \Lambda \times \Phi$.*

The entailment relation $\vdash$, introduced in Subsection 2.1, is extended to Λ as expected.

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 10, labels L are defined in Definition 12 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.*

Some of the rules in Table 2 are the same as those given in [1] for the pi-calculus. The most interesting among them 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. This implies, for example, that in rule (Par′) side condition $\mathrm{bn}(\alpha) \cap \mathrm{fn}(r) = \emptyset$ is *not necessary*, since δ shifts any variable in r to the right and, thus, x_0 does not appear in $\delta.\,r$.

The other rules of Table 2 are tailored to deal with fusions. Their aim is two-fold: to enable propagation and composition of fusions (e.g., rules (Exp), (Par$_1$), (Par$'_1$), and (Par$_f$)) and to close at each step with respect to fusions running in parallel (e.g. rules (Pre), (Pre′), (Fus)). Note that the side conditions ensure a saturation of process behaviours with respect to the observable fusions. Hence, for example, $x = y \mid y = k \mid p$ and $x = y \mid x = k \mid p$ have the same transitions. Rule (Pre′) might seem unusual: we will justify the need for this rule in example 1 below.

Proposition 3. *Let $\Delta = \langle \Sigma, L, R\rangle$ be the transition specification in Definition 13. Rules R are in De Simone format.*

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

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

$$\text{(Fus)}\quad \varphi.p \xrightarrow{\varphi'}_{\psi'} p|\psi \quad \varphi' \sqsubseteq \psi' \sqsubseteq \psi;\ \psi' \vdash \varphi = \varphi'$$

$$\text{(Exp)}\quad x = y \xrightarrow{\epsilon}_{x=y} x = y \qquad\qquad \text{(Sum)}\quad \frac{p \xrightarrow{\alpha}_{\varphi} p'}{p{+}q \xrightarrow{\alpha}_{\varphi} p'}$$

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

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

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

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

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

$$\text{(Par}_f\text{)}\quad \frac{p_1 \xrightarrow{\epsilon}_{\varphi_1} q_1 \quad p_2 \xrightarrow{\epsilon}_{\varphi_2} q_2}{p_1|p_2 \xrightarrow{\epsilon}_{\varphi'} q_1|q_2} \qquad \varphi' \sqsubseteq \varphi_1 + \varphi_2$$

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

$$\text{(Close)}\quad \frac{p_1 \xrightarrow{x}_{\delta\,(\varphi)} q_1 \quad p_2 \xrightarrow{\bar{x}}_{\delta\,(\varphi)} q_2}{p_1|p_2 \xrightarrow{\epsilon}_{\varphi} \nu.\, q_1|q_2}$$

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

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

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

Table 2. Structural Operational Semantics

Definition 14 (transition system lts_f).
The transition system for algebra B is $lts_g = \langle B, \rightarrow \rangle$, where $\rightarrow$ is defined by the SOS rules in Table 2 plus the following axiom:

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

Theorem 3. *Let B be the permutation algebra defined in Definition 10. Then, Condition 1 in Theorem 2 holds.*

Corollary 3. *Let B be the algebra defined in Definition 10. Bisimilarity is a congruence in $g : B \rightarrow P_{\Delta}(B)$.*

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

Hint of proof. The proof relies on the definition of three intermediate transition systems and their notions of bisimulation, which aim at modelling that fusion hyperequivalence is closed with respect to substitution.

The rules of the first transition system lts_1 are similar to those given in Table 1 for the fusion calculus. The only differences derive from the fact that here the restricted name is assumed to be always x_0. Hence, in case of bound actions, substitutions on the source of a transition must be reflected on its destination by restoring the extruded or fresh name to x_0.

For p a term of algebra B, we denote by $\texttt{Eq}(p)$ the equivalence relation obtained as the sum of all explicit fusions in p. Bisimulation on lts_1 is a relation $\mathcal{R}$ such that $p \,\mathcal{R}\, q$ implies:

1. $\texttt{Eq}(p) = \texttt{Eq}(q)$;
2. for each $p \xrightarrow{\alpha}_1 p'$ there is some $q \xrightarrow{\alpha'}_1 q'$ such that $\texttt{Eq}(p) \vdash \alpha = \alpha'$ 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 given in Def. 1. This fact can be proved by observing that $p \sim_1 q$ if and only if $\sigma(p) \sim_1 \sigma(q)$, for any substitutive effect σ of $\texttt{Eq}(p)$.

We then define transition system lts_2 by adding to lts_1 a rule for closing with respect to fusions in parallel:

$$\frac{p|\varphi \xrightarrow{\alpha}_1 q}{p \xrightarrow{\alpha,\varphi}_2 q} \tag{2}$$

Bisimulation and bisimilarity $\sim_2$ are analogous to those defined for lts_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 adding in parallel at each step any possible fusion (rule 2).

The third transition system lts_3 is obtained essentially by replacing rule 2 in lts_2 with:

$$\pi.p \xrightarrow{\pi,\,\varphi}_3 q|\varphi.$$

Bisimilarity $\sim_3$ is analogous to $\sim_2$ ($\longrightarrow_2$ replaces $\longrightarrow_3$). The proof of the theorem is concluded by showing that $\sim_3$ is equivalent to both $\sim_2$ and $\sim_g$.

Example 1. Consider two fusion agents $P = (y)\,(y = z.\,\bar{x}y.\,R)$ and $Q = \epsilon\,.\bar{x}z.R$, with $y \notin R$. As expected, P and Q are hyperequivalent. Let us now translate P and Q in terms of algebra B. For convenience, we abbreviate $\delta\,(w)$ with w_+, for any $w \in \mathfrak{N}$. Then, $[\![P]\!] = \nu.\ z_+ = x_0.\,\bar{x}_+x_0.\delta.\,[\![R]\!]$ and $[\![Q]\!] = \epsilon\,.\bar{x}z.[\![R]\!]$. By Theorem 4, $[\![P]\!] \sim_g [\![Q]\!]$. In fact, $[\![P]\!] \xrightarrow{\epsilon}_\epsilon \nu.\,\bar{x}_+x_0.\delta.\,[\![R]\!]\,|\,x_0 = z_+$; on the other side, $[\![Q]\!] \xrightarrow{\epsilon}_\epsilon \bar{x}z.[\![R]\!]$. By rule (Pre′), $\bar{x}z.[\![R]\!] \xrightarrow{\bar{x}}_\epsilon \delta.\,[\![R]\!]\,|\,x_0 = z_+$ and, thus, the analogous actions taken by $\nu.\,\bar{x}_+x_0.\delta.\,[\![R]\!]\,|\,x_0 = z_+$ can be simulated.

References

1. 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.
2. A. Corradini, M. Große-Rhode, R. Heckel. Structured transition systems as lax coalgebras. In *Proc. of CMCS'98*, ENTS 11. Elsevier Science, 1998.
3. A. Corradini, R. Heckel, and U. Montanari. Compositional SOS and beyond: A coalgebraic view of open systems. *Theoretical Computer Science* 280:163-192, 2002.
4. 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*.
5. P. Gardner, C. Laneve, and L. Wischik. The fusion machine (extended abstract). In *Proc. of CONCUR '02*, LNCS 2421. Springer-Verlag, 2002.
6. L. G. Meredith, S. Bjorg, and D. Richter. Highwire Language Specification Version 1.0. Unpublished manuscript.
7. U. Montanari and M. Pistore. Pi-Calculus, Structured Coalgebras and Minimal HD-Automata. In *Proc. of MFCS'00*, LNCS 1983. Springer, 2000.
8. 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*.
9. J. Parrow and B. Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. In *Proc. of LICS'98*. IEEE Computer Society Press, 1998.
10. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science* 249(1):3–80, 2000.
11. D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. of LICS'97*, IEEE. Computer Society Press, 1997.
12. B. Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. PhD thesis, Department of Computer Systems, Uppsala University, 1998.
13. L. Wischik and P. Gardner. Strong Bisimulation for the Explicit Fusion Calculus. To appear in *Proc. of FoSSaCS '04*.