Available online at www.sciencedirect.com
ScienceDirect
The Journal of Logic and Algebraic Programming 72 (2007) 78–97
THE JOURNAL OF LOGIC AND ALGEBRAIC PROGRAMMING
www.elsevier.com/locate/jlap

A compositional coalgebraic model of fusion calculus ☆

Maria Grazia Buscemi [a,*], Ugo Montanari [b]

[a] IMT Lucca Institute for Advanced Studies, Via San Micheletto 3, I-55100 Lucca, Italy
[b] University of Pisa, Italy

Abstract

This paper is a further step in exploring the labelled transitions and bisimulations of fusion calculi. We follow a recent theory by the same authors and previously applied to the pi-calculus for lifting calculi with structural axioms to bialgebras and, thus, we provide a compositional model of the fusion calculus with explicit fusions. In such a model, the bisimilarity relation induced by the unique morphism to the final coalgebra coincides with fusion hyperequivalence and it is a congruence with respect to the operations of the calculus. The key novelty in our work is that we give an account of explicit fusions through labelled transitions. Interestingly enough, this approach allows to exploit for the fusion calculus essentially the same algebraic structure used for the pi-calculus.

Keywords: Process calculi; Algebraic/coalgebraic models

1. Introduction

The design of globally distributed systems is more and more centered around exchange of messages. The main advantages of message passing are its conceptual simplicity, minimal infrastructure requirements and its neutrality with respect to back-ends and platforms of services. Both pi-calculus [15] and fusion calculus [19] convey the idea of message passing in a distilled form and, thus, they seem promising candidates to model foundational aspects of such a paradigm.

The fusion calculus 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. A fusion is a name equivalence that allows to use interchangeably in a term all names in the same 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 [12], or forms of pattern matching between pairs of messages.

In the 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. The explicit fusion calculus [25,12] is a variant that aims at

☆ Research partially supported by the EU IST-FP6 16004 Integrated Project Sensoria.
* Corresponding author.
E-mail addresses: marzia.buscemi@imtlucca.it (M.G. Buscemi), ugo@di.unipi.it (U. Montanari).

1567-8326/$ - see front matter
doi:10.1016/j.jlap.2007.05.001

guaranteeing asynchronous broadcasting of name equivalences to the environment. 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 [1,20] 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. Coalgebraic models can suitably represent transition systems. A coalgebraic framework [21] 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, in particular, 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 [23] 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 pi-calculus and fusion 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 [16,17]. 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 [4] 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 [4] to provide a bialgebraic model of fusion calculus with explicit fusions. In such a model, the bisimilarity relation induced by the unique morphism to the final coalgebra coincides with fusion hyperequivalence and it is a congruence with respect to the operations of the calculus. A key contribution of this work is to give an account of explicit fusions through labelled transitions which, to our knowledge, has previously been absent. We argue that this result does not only concern fusion calculi 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 [4] is compositional only with respect to parallel composition and restriction; constants are introduced to represent π-agents whose out-most operator is neither parallel composition nor restriction. Moreover, the theory in [4] does not apply to late and open π-calculus as this would require the introduction of arbitrary (i.e., possibly non-injective) name substitutions.

Our present model of the fusion calculus, on the other side, is fully compositional with respect to the operations of the calculus. This is accomplished by the introduction of explicit fusions into the underlying algebra. 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 [4] are satisfied. Remarkably enough, explicit fusions enable us to model substitutions within our theory, while keeping essentially the same permutation algebra considered in [4] 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 prove that the translation of fusion agents in our algebra is fully abstract with respect to Parrow and Victor hyperequivalence. As in [26], closure with respect to substitution is obtained by adding in parallel at each step any possible fusion.

Related work

Several approaches based on presheaves have been recently proposed as meta-models for nominal calculi (see [10, 9,13,14], among others). Presheaves are categories of functors from a given category to **Set**. In [10], for instance, presheaves are considered over the category **I** of finite sets and injective functions while, in the most general case of [14], they are taken over the category of relations and monotone functions. Our approach is analogous, since it relies on a category of algebras and algebras can be seen as cartesian functors from certain cartesian categories (Lawvere theories) to **Set**. In particular, in [11], finitely supported permutation algebras have been proved equivalent to sheaves over **I**. On the coalgebraic side, our aim is simpler than [10], since fusion hyperequivalence is congruence, while in the pi-calculus (early and late) bisimilarity is not a congruence and its congruence closure is not a bisimulation. However, as far as we know, explicit fusions have not been specifically treated in any categorical setting. Mousavi and Reiners [18] have proposed a syntactic format of structural axioms that guarantees bisimulation to be a congruence. Testing our 'bisimulation' condition on axioms is more involving than performing their syntactic check. The format required by [18], though, is too restrictive for us as, for instance, standard axioms like those for commutativity and associativity are ruled out.

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. A four-page abstract [5] of this work focuses on a fragment of the fusion calculus without replication and restriction.

2. Background

2.1. Names, fusion and permutations

We need some basic definitions and properties on names, fusions and permutations of names. We denote by $\mathfrak{N} = \{x_0, x_1, x_2, \ldots\}$ the infinite, countable, totally ordered set of *names* and we use $x, y, z \ldots$ to denote names. For R a binary relation over $\mathfrak{N}$, by $R^\star$ we denote the reflexive, symmetric and transitive closure of R with respect to $\mathfrak{N}$.

Definition 2.1 (*fusions*). *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 $\{(x, y), (y, x)\}$.

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 substitutions and we denote with $[y_1 \mapsto x_1, \ldots, y_n \mapsto x_n]$ 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 $[y/x, x/y]$. 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, except for x_0 which is not affected.

2.2. The fusion calculus

In this section we give an overview of the fusion calculus, which has been introduced in [19]. Here we consider a *monadic* version of the calculus.

The fusion calculus *agent terms*, ranged over by P, Q, ..., 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,[1] and *prefixes*, ranged over by π, are I/O actions or fusions:

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

The occurrences of x in $(x)\,P$ are bound and fusion effects with respect to x are limited to P; also, in $\mathtt{rec}\,X.\,P$ the occurrences of the *agent variable* X are bound in P. *Free names* and *bound names* of agent term P are defined as usual and we denote them with $\mathrm{fn}(P)$ and $\mathrm{bn}(P)$, respectively. 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. Two agent terms are alpha equivalent if they only differ by a change of bound names.

Definition 2.2. We define fusion agents as fusion calculus agent terms up to structural congruence $\equiv$, that is the least congruence satisfying the following axioms:

(alpha)	$P \equiv Q$ if P and Q are alpha equivalent
(par)	$P \,\|\, \mathbf{0} \equiv P \qquad P \,\|\, Q \equiv Q \,\|\, P \qquad P \,\|\, (Q \,\|\, R) \equiv (P \,\|\, Q) \,\|\, R$
(res)	$(x)\,\mathbf{0} \equiv \mathbf{0} \qquad (x)\,(y)\,P \equiv (y)\,(x)\,P$
	$(x)\,(P + Q) \equiv (x)\,P + (x)\,Q$
(scope)	$P \,\|\, (z)\,Q \equiv (z)\,(P \,\|\, Q)$ where $z \notin \mathrm{fn}(P)$

We denote by $\mathcal{F}$ the set of all fusion agents.

Note that the axiom "$(x)\,P \equiv P$ if $x \notin \mathrm{fn}(P)$" can be derived.

Definition 2.3. 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.

Definition 2.4. The family of transitions $P \overset{\gamma}{\longmapsto} Q$ 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 it is regulated by the scope of the binder (x).

Example 2.5. The interaction between the agents $\overline{u}v.P$ and $ux.Q$ results in a fusion of v and x. This fusion also affects any further process R running in parallel, as illustrated below:

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

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

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

[1] Recursion is guarded in P iff in every subterm $\mathtt{rec}\,X.\,Q$ of P, variable X appears within a context $\pi.\_$.

Table 1
LTS for fusion

(F-PRE) $\pi.P \overset{\pi}{\longmapsto} P$	(F-PAR) $\dfrac{P \overset{\gamma}{\longmapsto} Q}{P \mid R \overset{\gamma}{\longmapsto} Q \mid R} \, if\ \mathrm{bn}(\gamma) \cap \mathrm{fn}(R) = \emptyset$
(F-COM) $\dfrac{P \overset{\bar{x}y}{\longmapsto} P' \quad Q \overset{xz}{\longmapsto} Q'}{P \mid Q \overset{\{y=z\}}{\longmapsto} P' \mid Q'}$	(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) $\dfrac{P \overset{az}{\longmapsto} Q \quad a \notin \{z, \bar{z}\}}{(z)\,P \overset{a(z)}{\longmapsto} Q}$	(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) $\dfrac{P \equiv P' \quad P' \overset{\gamma}{\longmapsto} Q' \quad Q' \equiv Q}{P \overset{\gamma}{\longmapsto} Q}$

Definition 2.6 (*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 $\sigma(P')\ \mathcal{S}\ \sigma(Q')$ for some fusion agent Q' and substitutive effect σ of γ.
> P is bisimilar to Q, written $P \dot{\sim} Q$, if $P\ \mathcal{S}\ Q$ for some fusion bisimulation $\mathcal{S}$.

Definition 2.7 (*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 $op^B(h(a_1), \ldots, h(a_n)) = h(op^A(a_1, \ldots, a_n))$. 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 2.8 (*transition systems*). Let $\Gamma = \langle \Sigma, E \rangle$ be an algebraic specification, and L be a set of labels. A (*labelled*) *transition system* 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 relation. For $\langle p, l, q \rangle \in \longrightarrow_{lts}$ we write $p \overset{l}{\longrightarrow}_{lts} q$.

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 $p \overset{l}{\longrightarrow}_{lts} q$ implies $f(p) \overset{l}{\longrightarrow}_{lts'} f(q)$.

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

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 2.9 (*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 $i = 1, \ldots, n$ 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[2] to $p_i \xrightarrow{l_i} q_i$ for $i = 1, \ldots, n$ is also a solution to $p \xrightarrow{l} q$.

Definition 2.10 (*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 transition systems (i.e., transition systems whose states do not have an algebraic structure) can be represented as coalgebras for a suitable functor [21].

Definition 2.11 (*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 2.12. *For a fixed set of labels* L*, let* $P_L : \mathbf{Set} \to \mathbf{Set}$ *be the co-pointed functor defined on objects as* $P_L(X) = X \cup \mathcal{P}(L \times X)$*, 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 (co-pointed)* P_L*-coal gebras* f *such that*

$$f(x) \cap X = \{x\}, \quad \forall x \in X \tag{2}$$

are in a one-to-one correspondence with transition systems[3] *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* $\langle l, q \rangle \in f(p)$*.*

Definition 2.13 (*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 *(unary) 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 [23] 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

[2] 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 $\hat{h}(p) \xrightarrow{l}_{lts} \hat{h}(q)$.

[3] 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 2.8.

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 2.14 (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:*

- $|P_\Delta(A)| = P_L(|A|)$;
- *If $t \in A$ then $t \in P_\Delta(A)$;*
- *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*
  $$\frac{\langle l_i, p_i \rangle \in S_i,\ i \in I \quad q_j \in S_j,\ q_j \text{ is not a pair},\ j \notin I}{\langle l, p[p_i/y_i,\ i \in I,\ \ q_j/y_j,\ j \notin I] \rangle \in op^{P_\Delta(A)}(S_1, \ldots, S_n)};$$
- *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 2.15. *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 [4] and generalises the theory described so far to transition systems with structural axioms.

Theorem 2.16 (initial *lts*). *Let $\Delta = \langle \Gamma = \langle \Sigma, E \rangle, L, R \rangle$ be a transition specification with rules R in De Simone format, B be a Γ-algebra and $\mathrm{LTS}_{\Delta,B}$ be a class of transition systems $lts_{\Delta,B} = \langle B, \longmapsto_{lts_{\Delta,B}} \rangle$ over Δ.*

Then, there exists an initial transition system $\mathrm{lts}_{\Delta,B}$ in $\mathrm{LTS}_{\Delta,B}$ such that for all $lts_{\Delta,B}$ in $\mathrm{LTS}_{\Delta,B}$ and for all $p \in B$, $p \overset{l}{\longmapsto}_{\mathrm{lts}_{\Delta,B}} q$ implies $p \overset{l}{\longmapsto}_{lts_{\Delta,B}} q$.

Furthermore, the transitions of $\mathrm{lts}_{\Delta,B}$ can be derived using the rules R and the following additional rule:

$$\text{(Struct)}\ \frac{t_1 =_E t_1' \quad t_1' \overset{l}{\longmapsto}_{\mathrm{lts}_{\Delta,B}} t_2' \quad t_2' =_E t_2}{t_1 \overset{l}{\longmapsto}_{\mathrm{lts}_{\Delta,B}} t_2}$$

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

Theorem 2.17. *Let $\Delta = \langle \Gamma = \langle \Sigma, E \rangle, L, R \rangle$ be a transition specification with rules R in De Simone format, B be a Γ-algebra and $g_{\mathrm{lts}_{\Delta,B}}$ be the coalgebra associated to the initial transition system $\mathrm{lts}_{\Delta,B}$, as specified by Theorem 2.16. 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.*

Then, the coalgebra f satisfies the condition 2 of Proposition 2.12.

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):

$$\frac{x_i \xrightarrow{l_i} y_i \quad i \in I}{t_1 \xrightarrow{l} t_1'} \quad \textit{implies} \quad \frac{x_i \xrightarrow{l_i} y_i \quad i \in I}{t_2 \xrightarrow{l} t_2'} \quad \textit{and} \quad t_1' =_E t_2' \tag{3}$$

and viceversa, only using the rules in R. Then, $g_{\mathrm{lts}_{\Delta,B}}$ can be lifted to be a coalgebra on $\mathbf{Alg}(\Sigma)$. Moreover, bisimilarity on $\mathrm{lts}_{\Delta,B}$ is a congruence.

Proof (*Hint*). The proof mainly consists in proving that $=_E$ is a bisimulation. The thesis then follows by noting that h is surjective and by the fact that f, h, and $P_L(h)$ can be lifted from **Set** to **Alg**(Σ) and, thus, so can $g_{\mathrm{lts}_{\Delta,B}}$. □

3. A transition system for fusion calculus

In this section, following the approach adopted in [4] for the pi-calculus, we provide a transition system lts_F for the fusion calculus and apply the general result recalled in Section 2.3 to lift lts_F to a bialgebra.

We first define a permutation algebra [4,16] enriched with the operations of fusion calculus and with constants modelling explicit fusions $x = y$. The introduction of explicit fusions is intended to model substitutive effects of fusion calculus while keeping essentially the same permutation algebra as in [4]. 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 $x = y \mid \_$. 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 3.1. A signature Σ_F for the fusion calculus is defined as follows:

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

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

We adopt the convention that operators have decreasing binding power, in the following order: π, ρ, $|$, ν and δ. Thus, for example, $\nu.\,\delta.\,\rho p \,|\, \pi.q$ means $\nu.\,(\,\delta.\,(\,(\rho p) \,|\, (\pi.q)\,)\,)$.

Restriction ν corresponds to (x) in fusion calculus. The argument of ν is omitted because we assume that the extruded or restricted name in ν is always x_0. In fact, this idea resembles de Bruijn indexes, where the innermost bound variable is always denoted by index 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 also contains a constant $c_{\mathtt{rec}\,X.\,P}$ for each fusion agent $\mathtt{rec}\,X.\,P$. Further on, we will equip every constant with a set of operational rules that mimic any possible transition the corresponding agent can perform.

Definition 3.2 (*permutation algebra for fusion calculus*). A permutation algebra B_F for fusion calculus is the initial algebra $B_F = T_{\Sigma_F,E_F}$ where:

- Σ_F is the signature defined above;
- E_F is the set of axioms below:

(group) $(\rho' \circ \rho)p \doteq \rho'(\rho p)$ $\quad$ $\mathrm{id}\,p \doteq p$

(par) $p \,|\, \mathbf{0} \doteq p$ $\quad$ $p \,|\, q \doteq q \,|\, p$ $\quad$ $p \,|\, (q \,|\, r) \doteq (p \,|\, q) \,|\, r$

(res) $\nu.\,\mathbf{0} \doteq \mathbf{0}$ $\quad$ $\nu.\,((\delta.\,p) \,|\, q) \doteq p \,|\, \nu.\,q$ $\quad$ $\nu.\,\nu.\,[x_0 \leftrightarrow x_1]p \doteq \nu.\,\nu.\,p$

(perm) $\rho\,\mathbf{0} \doteq \mathbf{0}$ $\quad$ $\rho(\pi.p) \doteq \rho(\pi).\rho p$ $\quad$ $\rho(p \,|\, q) \doteq \rho p \mid \rho q$

$\rho\nu.\,p \doteq \nu.\,\rho_{+1}p$ $\quad$ $\rho\, c_{\mathtt{rec}\,X.\,P} \doteq c_{\rho(\mathtt{rec}\,X.\,P)}$

(delta) $\delta.\,\mathbf{0} \doteq \mathbf{0}$ $\quad$ $\delta.\,(\pi.p) \doteq \delta\,(\pi).\delta.\,p$ $\quad$ $\delta.\,p \,|\, q \doteq (\delta.\,p) \,|\, \delta.\,q$

$\delta.\,\nu.\,p \doteq \nu.\,[x_0 \leftrightarrow x_1]\delta.\,p$ $\quad$ $\delta.\,\rho p \doteq \rho_{+1}\delta.\,p$

$\delta.\; c_{\mathtt{rec}\,X.\,P} \doteq c_{\delta\,(\mathtt{rec}\,X.\,P)}$

(fus) $x = x \doteq \mathbf{0}$ $\quad$ $\nu.\,(x_0 = x) \doteq \mathbf{0}$ $\quad$ $\rho(x = y) \;\doteq\; \rho(x) = \rho(y)$

$\delta.\,x = y \;\doteq\; \delta\,(x) = \delta\,(y)$

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 and, in particular, the second (**res**) rule is the counterpart of the scope extrusion law. 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. The 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.

Note that the above axioms can be applied from left to right to reduce every term p into a canonical form in which ρ and δ only occur as syntactically applied to explicit fusions and prefixes. For example, a term $p = \rho\big(\nu.\,x_2x_0.\varphi.\mathbf{0}\,|\,\delta.\,\overline{x}_2x_3.\mathbf{0}\,|\,x_3 = x_4\,|\,c_{\mathtt{rec}\,X.\,Q}\big)$, with $\rho = [x_1 \leftrightarrow x_4]$, can be reduced to the normal form $p' = \nu.\,x_5x_0.\rho(\varphi).\mathbf{0}\,|\,\overline{x}_3x_1.\mathbf{0}\,|\,x_3 = x_1\,|\,c_{\rho(\mathtt{rec}\,X.\,Q)}\big)$.

We give below a translation of fusion agents into terms of algebra B_F. Then, we define a transition system *lts* for the algebra B_F and show that axioms in E_F satisfy the bisimilarity condition for lifting coalgebras to bialgebras, as required by Theorem 2.17.

Definition 3.3 (*translation* $[\![\cdot]\!]$). We define a translation of fusion agents $[\![\cdot]\!] : \mathcal{F} \to |B_F|$ as follows:

$$[\![\mathbf{0}]\!] = \mathbf{0} \qquad [\![\pi.P]\!] = \pi.[\![P]\!] \qquad [\![P\,|\,Q]\!] = [\![P]\!]\,|\,[\![Q]\!]$$

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

The translation is straightforward, except for restriction ν that gives the flavour of the De Bruijn 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. For example, the translation of a fusion agent $P = (x_2)\,(\{x_2 = x_4\}.\,\bar{x}_7x_2.\,\mathbf{0})$ is $[\![P]\!] = \nu.\,[x_0 \leftrightarrow x_3]\delta.\,\{x_2 = x_4\}.\,\bar{x}_7x_2.\mathbf{0} = \nu.\,[x_0 \leftrightarrow x_3]\{x_3 = x_5\}.\,\bar{x}_8x_3.\mathbf{0} = \nu.\,\{x_0 = x_5\}.\,\bar{x}_8x_0.\mathbf{0}$.

Theorem 3.4. *Let P and Q be two fusion agents. If $P \equiv Q$ then $[\![P]\!] \doteq [\![Q]\!]$.*

Theorem 3.5. *Let $[\![\cdot]\!]$ be the translation defined in Def. 3.3 and let $\underline{B}_F$ be the algebra B_F without explicit fusions. Let $\{\!\!\{\cdot\}\!\!\} : |\underline{B}_F| \to \Pi$ be a translation defined as follows: $\{\!\!\{\mathbf{0}\}\!\!\} = \mathbf{0}$; $\{\!\!\{\pi.p\}\!\!\} = \pi.\{\!\!\{p\}\!\!\}$; $\{\!\!\{p|q\}\!\!\} = \{\!\!\{p\}\!\!\}|\{\!\!\{q\}\!\!\}$; $\{\!\!\{\nu.\,p\}\!\!\} = (\boldsymbol{\nu} x_i)\ \nu\,(\,)[\delta\,(x_i) \leftrightarrow x_0]\{\!\!\{p\}\!\!\}$, where x_i is chosen such that $\delta\,(x_i) \notin \mathrm{fn}(\{\!\!\{p\}\!\!\})$; $\{\!\!\{c_{\mathtt{rec}\,X.\,P}\}\!\!\} = \mathtt{rec}\,X.\,P$; $\{\!\!\{\rho p\}\!\!\} = \rho(\{\!\!\{p\}\!\!\})$; $\{\!\!\{\delta.\,p\}\!\!\} = \delta\,(\{\!\!\{p\}\!\!\})$. Then, for every fusion agent P, $\{\!\!\{[\![P]\!]\}\!\!\} \equiv P$, and for every term p in $\bar{B}_F$, $[\![\{\!\!\{p\}\!\!\}]\!] \doteq p$.*

Definition 3.6. We let L_F be the set of labels $L_F = \Lambda \times \Phi$, where $\Lambda = \{xy,\ \overline{x}\,y,\ x,\ \overline{x},\ \varphi,\ -\mid x, y, \mathrm{n}(\varphi) \in \mathfrak{N}\}$ and $-$ denotes a *null action*, and Φ is the set of all fusions over $\mathfrak{N}$. We let $\alpha, \beta, \ldots$ range over Λ.

The correspondence between the left components of the labels L_F and the actions of the fusion calculus is the obvious one for xy, $\overline{x}\,y$, φ. Bound input and bound output are denoted by x and $\overline{x}$, respectively, where the bound name is implicitly assumed to be x_0. Unlike standard actions in transition systems, the null action does not represent any observation that the environment can perform on a process. The reason for introducing the null action is mainly technical. In fact, we include explicit fusions among the terms of our algebra although they behave differently from standard processes. Thus, as we will see below, we need to introduce special transitions like

$$x = y \xrightarrow{(-,\,x=y)} x = y$$

that are not intended to model the evolution of a system but rather to specify the *unifications* that are available to the processes running in parallel.

By $\rho(\alpha)$, for $\alpha \neq -$, we denote the syntactic application of ρ to the names of α; by $\delta\,(\alpha)$ and $\nu\,(\alpha)$, for $\alpha \neq -$, 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. Moreover, $\rho(-) = \delta\,(-) = \nu\,(-) = -$. For instance, $[x_3 \leftrightarrow x_5]\bar{x}_5x_1 = \bar{x}_3x_1$, $\delta\,(\overline{x}_1x_3) = \overline{x}_2x_4$ and, for $\varphi = \{x_0 = x_2 = x_5\}$, $\nu\,(\varphi) = \{x_1 = x_4\}$.

The following notion of *entailment relation* will be useful in the definition of the operational rules below.

Definition 3.7. The *entailment relation* $\vdash$ is defined as follows: $\varphi \vdash \alpha = \beta$ if α, β are not fusions and $\sigma(\alpha) = \sigma(\beta)$, for a substitutive effect σ of φ; $\varphi \vdash \alpha = \beta$ if α, β are fusions and $\varphi + \alpha = \varphi + \beta$.

Table 2
Basic SOS rules

(Rho)	$\dfrac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha \neq x, \bar{x}}{\rho p \xrightarrow{(\rho(\alpha),\, \rho(\varphi))} \rho q}$	(Rho′)	$\dfrac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha = \bar{x}, x}{\rho p \xrightarrow{(\rho_{+1}(\alpha),\, \rho_{+1}(\varphi))} \rho_{+1} q}$
(Del)	$\dfrac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha \neq x, \bar{x}}{\delta.\, p \xrightarrow{(\delta(\alpha),\, \delta(\varphi))} \delta.\, q}$	(Del′)	$\dfrac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha = \bar{x}, x}{\delta.\, p \xrightarrow{(\delta(\alpha),\, \delta(\varphi))} [x_0 \leftrightarrow x_1]\delta.\, q}$
(Par)	$\dfrac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha \neq x, \bar{x}, -}{p \mid r \xrightarrow{(\alpha,\varphi)} q \mid r}$	(Par′)	$\dfrac{p \xrightarrow{(\alpha,\varphi)} q \quad \alpha = x, \bar{x}}{p \mid r \xrightarrow{(\alpha,\varphi)} q \mid \delta.\, r}$
(Res)	$\dfrac{p \xrightarrow{(\delta(\alpha),\, \varphi)} q \quad \alpha = \bar{x}y, xy, -}{\nu.\, p \xrightarrow{(\alpha,\, \nu(\varphi))} \nu.\, q}$	(Res′)	$\dfrac{p \xrightarrow{(\delta(\alpha),\, \varphi)} q \quad \alpha = \bar{x}, x}{\nu.\, p \xrightarrow{(\alpha,\, \nu(\varphi))} \nu.\, [x_0 \leftrightarrow x_1]q}$
(Open)	$\dfrac{p \xrightarrow{(xx_0,\, \delta(\varphi))} q \quad x \neq x_0}{\nu.\, p \xrightarrow{(x,\, \delta(\varphi))} q}$	(Scope)	$\dfrac{p \xrightarrow{(\varphi,\psi)} q}{\nu.\, p \xrightarrow{(\nu(\varphi),\, \nu(\psi))} \nu.\, q}$
(Com)	$\dfrac{p_1 \xrightarrow{(xy,\varphi)} q_1 \quad p_2 \xrightarrow{(\bar{x}z,\varphi)} q_2}{p_1 \mid p_2 \xrightarrow{(y=z,\, \varphi)} q_1 \mid q_2 \mid y = z}$	(Com′)	$\dfrac{p_1 \xrightarrow{(xy,\varphi)} q_1 \quad p_2 \xrightarrow{(\delta(\bar{x}),\, \delta(\varphi))} q_2}{p_1 \mid p_2 \xrightarrow{(\tau,\varphi)} q_1 \mid \nu.\, (q_2 \mid \delta(y) = x_0)}$
(Close)	$\dfrac{p_1 \xrightarrow{(x,\varphi)} q_1 \quad p_2 \xrightarrow{(\bar{x},\varphi)} q_2}{p_1 \mid p_2 \xrightarrow{(\tau,\, \nu(\varphi))} \nu.\, (q_1 \mid q_2)}$	(Rec)	$\dfrac{[\![P[\mathtt{rec}\, X.\, P \,/\, X]]\!] \xrightarrow{(\alpha,\varphi)} q}{c_{\mathtt{rec}\, X.\, P} \xrightarrow{(\alpha,\varphi)} q}$

There is also a rule analogous to Rule (Open) but with output actions; rule (Com′) has a symmetric counterpart.

Definition 3.8 (*transition specification* Δ_F). The transition specification Δ_F is the tuple $\langle \Sigma_F, L_F, R_F \rangle$, where the signature Σ_F is as in Definition 3.2, labels L_F are defined in Definition 3.6 and R_F is the set of SOS rules in Tables 2, 3 and 4. Transitions take the form $p \xrightarrow{(\alpha,\varphi)} q$, where (α, φ) ranges over L.

The rules in Table 2 concern permutations, restriction, parallel composition and replication, and are essentially the same as those given in [4] for the pi-calculus. The most interesting among them are those with bound I/O actions ((Rho′), (Del′), (Par′), (Res′), and (Com′)): they follow the intuition that substitutions on the source of a transition must be reflected on its target by restoring the extruded or fresh name to x_0. Thus, for example, rule (Del′) 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 $\delta.\, r$. Note that neither (Par) nor (Par′) are applied with $\alpha = -$. In fact, the transitions performed by the combination of explicit fusions with each other and with other terms are regulated by the rules in Table 3.

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 [4] for the pi-calculus.

The rules in Tables 3 are suited to deal with explicit fusions. Transitions of the form $p \xrightarrow{(-,\varphi)} q$ are special transitions, which are meant to express that two names in the same equivalence class of φ can be used interchangeably in p. 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 running in parallel. Rule (Nil) is justified by the fact that we expect $\nu.\, (x = x_0)$ be bisimilar to $\mathbf{0}$, in accordance with the semantics of the explicit fusion calculus. In fact, since neither in $\nu.\, (x = x_0)$ nor in $\mathbf{0}$ there is a pair of names that can be used one for the other, both the terms only have a transition $\xrightarrow{(-,\tau)}$.

According to the rules in Table 4, a term can perform a non-null action as long as the action in the label can be 'unified' with the prefix; the unification φ is included in the target term and any unification φ' which is smaller than φ

Table 3
SOS rules for explicit fusions

(Exp) $x = y \xrightarrow{(-,\varphi)} x = y \quad x \neq y;\ \varphi \sqsubseteq x = y$	(Nil) $\mathbf{0} \xrightarrow{(-,\tau)} \mathbf{0}$
(Par$_1$) $\dfrac{p_1 \xrightarrow{(\alpha,\varphi_1)} q_1 \quad p_2 \xrightarrow{(-,\varphi_2)} q_2 \quad \alpha = \bar{x}y, xy, \varphi}{p_1 \mid p_2 \xrightarrow{(\beta,\varphi')} q_1 \mid q_2}\ \varphi' \sqsubseteq \varphi_1 + \varphi_2;\ \varphi_1 + \varphi_2 \vdash \alpha = \beta$	
(Par$'_1$) $\dfrac{p_1 \xrightarrow{(\alpha,\varphi_1)} q_1 \quad p_2 \xrightarrow{(-,\varphi_2)} q_2 \quad \alpha = \bar{x}, x}{p_1 \mid p_2 \xrightarrow{(\beta,\varphi')} q_1 \mid \delta . q_2}\ \varphi' \sqsubseteq \varphi_1 + \delta(\varphi_2);\ \varphi_1 + \delta(\varphi_2) \vdash \alpha = \beta$	
(Par$_f$) $\dfrac{p_1 \xrightarrow{(-,\varphi_1)} q_1 \quad p_2 \xrightarrow{(-,\varphi_2)} q_2}{p_1 \mid p_2 \xrightarrow{(-,\varphi')} q_1 \mid q_2}\ \varphi' \sqsubseteq \varphi_1 + \varphi_2$	

Table 4
SOS rules for closure wrt fusion contexts

(Pre) $xy.p \xrightarrow{(x'y',\varphi')} p \mid \varphi$	$\varphi' \sqsubseteq \varphi;\ \varphi \vdash xy = x'y'$
(Pre$'$) $xy.p \xrightarrow{(\delta(x'),\delta(\varphi'))} (\delta.(p \mid \varphi)) \mid x_0 = \delta(y)$	$\varphi' \sqsubseteq \varphi;\ \varphi \vdash x = x'$
(Fus) $\varphi.p \xrightarrow{(\varphi',\psi')} p \mid \psi + \varphi$	$\psi' \sqsubseteq \psi;\ \psi \vdash \varphi = \varphi'$

Rules (Pre), (Pre$'$) are analogous with output actions.

is observed as a 'side-effect '. For example, in rule (Pre) the prefix xy and the action $x'y'$ must be identified by a name fusion φ ($\varphi \vdash xy = x'y'$) and the right-hand component of the label is any fusion φ' such that $\varphi' \sqsubseteq \varphi$.

As we will see below, the rules in Table 4 guarantee that the associated bisimilarity is preserved by closure with respect to fusions running in parallel. As an example, consider the term $p = \overline{x}\, y.y\, w.\mathbf{0}$. By rule (Pre) p can take any of the following steps:

$$p \xrightarrow{(\overline{x}\, y,\tau)} y\, w.\mathbf{0} \qquad p \xrightarrow{(\overline{z}\, y,\tau)} y\, w.\mathbf{0} \mid z = x \qquad p \xrightarrow{(\overline{x}'y',\psi)} y\, w.\mathbf{0} \mid \varphi$$

for all φ, for all x', y' such that $\varphi \vdash xy = x'y'$, and for all ψ such that $\psi \sqsubseteq \varphi$.

The presence of two rules for I/O prefixes, (Pre) and (Pre$'$), is required in order to have a fully abstract translation of fusion agents into terms of the algebra, as shown in the following Example 3.9. Note that all side conditions of the rules in Tables 3 and 4 are meant to ensure closure of process behaviours with respect to the explicit fusions. This form of saturation is formalised in Proposition 3.12. Note that the rules in Tables 2, 3 and 4 yield an infinite branching transition system.

Example 3.9. Consider two fusion agents $P = (x_2)\,(\{x_2 = x_4\}.\,\bar{x}_7 x_2.\,\mathbf{0})$ and $Q = \tau\,.\bar{x}_7 x_4.\,\mathbf{0}$. Of course, P and Q are hyperequivalent. Let us now translate P and Q in terms of algebra B_F: $[\![P]\!] = \nu.\ \{x_0 = x_5\}.\,\bar{x}_8 x_0.\mathbf{0}$ and $[\![Q]\!] = \tau\,.\bar{x}_7 x_4.\mathbf{0}$. The terms $[\![P]\!]$ and $[\![Q]\!]$ have the same transitions. Indeed, the only possible transitions of $[\![P]\!]$ are in the form $P \xrightarrow{(\tau,\psi')} \nu.\,(\bar{x}_8 x_0.\mathbf{0} \mid x_0 = x_8) \mid \psi$, for any ψ and ψ' such that $\psi' \sqsubseteq \psi$; thus, they can be mimicked by $[\![Q]\!] \xrightarrow{(\tau,\psi')} \bar{x}_7 x_4.\mathbf{0} \mid \psi$. Next, suppose $\nu.\,(\bar{x}_8 x_0.\,\mathbf{0} \mid x_0 = x_8) \mid \psi \xrightarrow{(\bar{x}_8,\varphi')} x_0 = x_5 \mid \psi \mid \varphi$ with $\varphi' \sqsubseteq \varphi$. By rule (Pre$'$), $\bar{x}_7 x_4.\mathbf{0}$ also have the same transition.

Proposition 3.10. *Rules R are in De Simone format.*

In virtue of the above proposition, the hypotheses of Theorem 2.16 are satisfied and, thus, there exists the initial transition system $\mathrm{lts}_{\Delta_F,B_F}$. In the following, we will abbreviate $\mathrm{lts}_{\Delta_F,B_F}$ by lts_F and its associated notion of bisimilarity $\sim_{\mathrm{lts}_{\Delta_F,B_F}}$ by $\sim_F$.

We remark that the bisimilarity $\sim_F$ is very close in spirit to the *inside-outside* bisimulation defined in [25], which satisfies the following properties: (1) two equivalent terms must contain the same explicit fusions and (2) two equivalent names must behave in the same way under explicit fusions contexts.

We now introduce a notion of equivalence relation $\mathtt{Eq}(p)$, induced by the explicit fusions in a term p. The notation given for fusions in Def. 2.1 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 3.11. 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:

$$\begin{array}{lll}
\mathtt{Eq}(\mathbf{0}) = \tau & \mathtt{Eq}(\pi.p) = \tau & \mathtt{Eq}(p \mid q) = \mathtt{Eq}(p) + \mathtt{Eq}(q) \\
\mathtt{Eq}(\nu.\, p) = \nu\,(\mathtt{Eq}(p)) & \mathtt{Eq}(\rho p) = \rho(\mathtt{Eq}(p)) & \mathtt{Eq}(\delta.\, p) = \delta\,(\mathtt{Eq}(p)) \\
\mathtt{Eq}(x = y) = \{x = y\} & \mathtt{Eq}(c_{\mathtt{rec}\,X.\,P}) = \tau &
\end{array}$$

For example, for $p = (x = y) \mid (y = z) \mid p'$, $\mathtt{Eq}(p) = \{x = y = z\} + \mathtt{Eq}(p')$ while, for $q = x_3 x_2.\mathbf{0} \mid \nu.\,(x_4 = x_0 \mid x_4 = x_6)$, $\mathtt{Eq}(q) = x_3 = x_5$.

Proposition 3.12

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 3.13

- The terms $p_1 = (x = y) \mid (y = k) \mid p$ and $p_2 = (x = y) \mid (x = k) \mid p$ 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 $\varphi = y = k$.
- Let $p = \bar{x}y.p_1 \mid zk.p_2$. By rules (Pre) and (Com), $p \xrightarrow{(y=k,\,\varphi)} p_1 \mid p_2 \mid \psi \mid y = k$, 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.

Theorem 3.14. *If we consider* Δ_F *as transition specification* Δ, B_F *as algebra* B *and* lts_F *as initial transition system then Condition* 3 *in Theorem* 2.17 *holds.*

Proof. See the appendix. □

Corollary 3.15. *The coalgebra* g_{lts_F} *can be lifted to be a bialgebra in* $\mathbf{Alg}(\Sigma_F)$ *and, thus, bisimilarity in* lts_F *is a congruence.*

Proof. It follows by Theorems 3.14 and 2.17. □

Our next claim is that the translation $[\![\cdot]\!]$ of fusion agents into terms of the permutation algebra B_F 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 3.16. *Let* P *and* Q *be two fusion agents. Then,* $P \sim_{he} Q$ *iff* $[\![P]\!] \sim_F [\![Q]\!]$.

Proof (*Hint*). The proof relies on the definition of three intermediate transition systems and their notions of bisimulation, which aim at providing a notion of equivalence closed with respect to substitutions, akin to fusion hyperequivalence. We briefly introduce the transition systems and the associated equivalence relations; the main features of each of them are summarised in Table 5.

The first transition system lts_1 is defined by the rules given in Table B.1. The first group of rules is similar to those in Table 2, while rules for prefixes ((Pre), (Pre′), and (Fus)) do not consider all the possible fusions running in parallel and, thus, they differ from those given in Table 4. In fact, the aim of lts_1 is to ensure saturation of behaviours of a term

Table 5
Transition systems summary

	lts_1	lts_2	lts_3	lts_F
Fusion combination and propagation	(EQ)	(EQ)	(EQ)	(EXP), (NIL), (PAR$_1$), (PAR$'_1$), (PAR$_f$)
Closure wrt fusions in parallel	NO	(CTX)	(PRE), (PRE$'$), (FUS)	(PRE), (PRE$'$), (FUS)
Correspondence with fusion calculus	$\sim_1 = \dot\sim$	$\sim_2 = \sim_{he}$	$\sim_3 = \sim_{he}$	$\sim_F = \sim_{he}$

only with respect to the explicit fusions contained in the term, but lts_1 is not intended to guarantee that its associated bisimilarity is preserved by closure with respect to fusions contexts. Moreover, lts_1 contains a 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 in Table 3. The fact that (EQ) has the same effect of the rules in Table 3 easily follows by observing that by (EQ) lts_1 enjoys a special case of the saturation property (Proposition 3.12.1), disregarding φ. Thus, for example, $x = y \mid \bar{x}z.\mathbf{0}$ 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 terms 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 2.6.

Next, we define a second transition system lts_2 by adding to lts_1 a rule for closing with respect to fusions in parallel:

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

Bisimilarity $\sim_2$ is analogous to $\sim_1$ (with $\rightarrow_2$ in place of $\rightarrow_1$). 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 substitutions, by considering any possible fusion context (rule (CTX)).

The third transition system lts_3 is defined by the rules in Table 2 and 4 plus rule

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

Bisimilarity $\sim_3$ is analogous to $\sim_2$ ($\longrightarrow_3$ replaces $\rightarrow_2$).

The proof of the theorem is concluded by showing that $\sim_3$ is equivalent to both $\sim_2$ and $\sim_F$. 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 in Table 4.

Finally, the equivalence of $\sim_3$ and $\sim_F$ follows from the fact that lts_3 satisfies the saturation property stated in Proposition 3.12, by means of the combined use of rules in Table 4 and (EQ). □

4. Conclusions

In this paper, we have provided a bialgebraic model of the fusion calculus, such that the bisimilarity relation induced by the unique morphism to the final coalgebra coincides with fusion hyperequivalence and it is a congruence with respect to the operations of the calculus. Specifically, we have introduced a permutation algebra with the operations of the fusion calculus along with constants modelling explicit fusions and recursion. Subsequently, we have defined a transition system for the enriched permutation algebra and we have proved that the conditions required by the result presented in [4] for lifting calculi with structural axioms to bialgebras are satisfied. Unfortunately, the proposed transition system is infinite branching. We plan to develop an alternative symbolic transition system to cope with this problem.

We would also like to investigate whether our approach can be applied to the pi-calculus open bisimulation [22] and to generalisations of the fusion calculus, such as D-Fusion [2] and U-Calculus [3]. We argue that the theory developed in [4] cannot be straightforwardly applied, because these extended calculi exploit a notion of ‘distinction’ to express that two names in a process cannot be fused. Our proposal is to cast distinctions within our model following the approach developed in [6] for generalising name fusions to arbitrary constraints, and to extend the above theory with types, by defining an underlying multi-sorted permutation algebra, whose sorts are the distinctions. This is in line with the approach of [14], where distinctions are represented as objects of the category over which functors are taken.

We are also interested in studying the relation with other meta-models of nominal calculi and, in particular, with the approaches based on presheaf categories.

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 [4] be flexible enough to allow varying the underlying algebra while employing similar constructions.

Acknowledgments

We are grateful to an anonymous referee for very helpful comments.

Appendix

A. Proof of Theorem 3.14

The proof consists in showing that for each $t_1 \doteq t_2$ in E_F, 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 $t_1 \doteq t_2$.

Ax. $\nu.\,(\delta.\,p)\,|\,q \doteq p|\nu.\,q$. There are four possible cases.

(1) By rule (Res), $\nu.\,(\delta.\,p)|q \xrightarrow{(\alpha,\,\nu\,(\varphi))} \nu.\,p'$, with $\alpha = xy, \bar{x}y, -$. Necessarily, $(\delta.\,p)|q \xrightarrow{(\delta\,(\alpha),\,\varphi)} p'$ and there are the following possible cases.

(a) By rule (Par), suppose $\delta.\,p \xrightarrow{(\delta\,(\alpha),\,\varphi)} p''$ (similarly, otherwise) and $p' = p''|q$. Then, $p \xrightarrow{(\alpha,\,\nu\,(\varphi))} p'''$ and $p'' = \delta.\,p'''$.

On the other hand, by rule (Par), $p|\nu.\,q \xrightarrow{(\alpha,\,\nu\,(\varphi))} p'''|\nu.\,q$.

(b) By rule (Par$_1$) $\delta.\,p \xrightarrow{(\delta\,(\beta),\,\varphi_1)} p''$, $q \xrightarrow{(-,\,\varphi_2)} q'$, with $\varphi \sqsubseteq \varphi_1 + \varphi_2$ and $\varphi_1 + \varphi_2 \vdash \delta\,(\alpha) = \delta\,(\beta)$; moreover $p' = p''\,|\,q'$. Necessarily rule (Del) has been applied. Thus, $p \xrightarrow{(\beta,\,\nu\,(\varphi_1))} p'''$, with $p'' = \delta.\,p'''$. Now consider $p\,|\,\nu.\,q$. By rule (Res) $\nu.\,q \xrightarrow{(-,\,\nu\,(\varphi_2))} \nu.\,q'$ and, by rule (Par$_1$), $p\,|\,\nu.\,q \xrightarrow{(\alpha,\,\nu\,(\varphi))} p'''\,|\,\nu.\,q'$. And $\nu.\,p' \doteq p'''\,|\,\nu.\,q'$.

(2) By rule (Res′), $\nu.\,(\delta.\,p)|q \xrightarrow{(\alpha,\,\nu\,(\varphi))} \nu.\,[x_0 \leftrightarrow x_1]p'$, with $\alpha = x, \bar{x}$ and $(\delta.\,p)|q \xrightarrow{(\delta\,(\alpha),\,\varphi)} p'$. There are the following possible cases.

(a) By rule (Par′), suppose $\delta.\,p \xrightarrow{(\delta\,(\alpha),\,\varphi)} p''$ with $p' = p''|\delta.\,q$ (similarly, otherwise). By rule (Del′), $p \xrightarrow{(\alpha,\,\nu\,(\varphi))} p'''$ and $p'' = [x_0 \leftrightarrow x_1]\delta.\,p'''$.

On the other hand, by rule (Par′), $p|\nu.\,q \xrightarrow{(\alpha,\,\nu\,(\varphi))} p'''|\delta.\,\nu.\,q$. By axioms E_F, $\nu.\,[x_0 \leftrightarrow x_1]([x_0 \leftrightarrow x_1](\delta.\,p''')|\delta.\,q) \doteq p'''|\delta.\,\nu.\,q$.

(b) By rule (Par$'_1$), suppose $q \xrightarrow{(\delta\,(\beta),\,\varphi_1)} q'$ and $\delta.\,p \xrightarrow{(-,\,\varphi_2)} p''$ (similarly, otherwise), with $p' = \delta.\,p''\,|\,q'$, $\varphi \sqsubseteq \varphi_1 + \delta\,(\varphi_2)$, and $\varphi_1 + \delta\,(\varphi_2) \vdash \delta\,(\alpha) = \delta\,(\beta)$. By rule (Del), $p \xrightarrow{(-,\,\nu\,(\varphi_2))} p'''$, and $p'' = \delta.\,p'''$.

On the other hand, by rule (Res′), $\nu.\,q \xrightarrow{(\beta,\,\nu\,(\varphi_1))} \nu.\,[x_0 \leftrightarrow x_1]q'$. By rule (Par$'_1$), $p\,|\,\nu.\,q \xrightarrow{(\alpha,\,\nu\,(\varphi))} \delta.\,p'''\,|\,\nu.\,[x_0 \leftrightarrow x_1]q'$. By axioms E_F, $\nu.\,[x_0 \leftrightarrow x_1](\,(\delta.\,\delta.\,p''')\,)\,|\,q' \doteq (\delta.\,p''')\,|\,\nu.\,[x_0 \leftrightarrow x_1]q'$.

(3) By rule (Open), $\nu.\,(\delta.\,p)|q \xrightarrow{(\bar{x},\,\delta\,(\varphi))} p'$. Necessarily, $\delta.\,p|q \xrightarrow{(\bar{x}x_0,\,\delta\,(\varphi))} p'$ and $x \neq x_0$. There are two possible cases.

(a) By rule (Par), suppose $q \xrightarrow{(\bar{x}x_0,\,\delta\,(\varphi))} q'$ (similarly, otherwise) and $p' = \delta.\,p|q'$.

On the other hand, by rule (Open), $\nu.\,q \xrightarrow{(\bar{x},\,\delta\,(\varphi))} q'$ and, by rule (Par′), $p\,|\,\nu.\,q \xrightarrow{(\bar{x},\,\delta\,(\varphi))} (\delta.\,p)|q'$.

(b) By rule (Par$_1$), suppose $q \xrightarrow{(\bar{y}z,\,\varphi_1)} q'$, $\delta.\,p \xrightarrow{(-,\,\varphi_2)} p''$, with $\delta\,(\varphi) \sqsubseteq \varphi_1 + \varphi_2$ and $\varphi_1 + \varphi_2 \vdash \bar{x}x_0 = \bar{y}z$ and $p' = p''\,|\,q'$. Necessarily, by rule (Del), $p \xrightarrow{(-,\,\varphi_3)} p'''$, with $\varphi_2 = \delta\,(\varphi_3)$ and $p'' = \delta.\,p'''$. Thus, φ_2 does not substitute x_0 and so $\varphi_1 \vdash \bar{x}x_0 = \bar{y}z$. It follows that, by Proposition 3.12, $q \xrightarrow{(\bar{x}x_0,\, \varphi_4)} q'$ and, by rule (Open), $\nu.\, q \xrightarrow{(\bar{x},\, \varphi_4)} q'$, with $\varphi_1 = \delta\,(\varphi_4)$. By rule (Par$'_1$), $p \mid \nu.\, q \xrightarrow{(\bar{x},\, \delta\,(\varphi))} \delta.\, p''' \mid q'$.

(4) By rule (Scope), $\nu.\,(\delta.\, p \mid q) \xrightarrow{(\nu\,(\varphi),\, \nu\,(\psi))} \nu.\, p'$ and $\delta.\, p \mid q \xrightarrow{(\varphi,\, \psi)} p'$. The most interesting cases are as follows.

(a) By rule (Com), suppose $\delta.\, p \xrightarrow{(xy,\, \psi)} p''$ and $q \xrightarrow{(\bar{x}z,\, \psi)} q'$ (similarly, otherwise), with $\varphi = (y = z)$, $p' = p'' \mid q' \mid \varphi$. Also, necessarily, rule (Del) has been applied with $p \xrightarrow{(x'y',\, \nu\,(\psi))} p'''$ and $p'' = \delta.\, p'''$, $xy = \delta\,(x'y')$.

On the other hand, by rule (Res), with $x, z \neq x_0$, $\nu.\, q \xrightarrow{(\nu\,(\bar{x}z),\, \nu\,(\psi))} \nu.\, q'$. Then, by rule (Com), $p \mid \nu.\, q \xrightarrow{(\nu\,(\varphi),\, \nu\,(\psi))} p''' \mid \nu.\, q' \mid \nu\,(\varphi)$. The thesis follows by the fact that $\nu.\,(\delta.\, p''' \mid q' \mid \varphi) \doteq p''' \mid \nu.\, q' \mid \varphi$, since $x_0 \notin \mathrm{n}(\varphi)$.

(b) By rule (Par$_1$), suppose $\delta.\, p \xrightarrow{(\varphi',\, \psi_1)} p''$ and $q \xrightarrow{(-,\, \psi_2)} q'$ (similarly, otherwise), with $\psi \sqsubseteq \psi_1 + \psi_2$, $\psi_1 + \psi_2 \vdash \varphi' = \varphi$, and $p' = p'' \mid q'$. Also, necessarily, rule (Del) has been applied with $p \xrightarrow{(\nu\,(\varphi'),\, \nu\,(\psi_1))} p'''$ and $p'' = \delta.\, p'''$.

On the other hand, by rule (Res), $\nu.\, q \xrightarrow{(-,\, \nu\,(\psi_2))} \nu.\, q'$ and, by rule (Par$_1$), $p \mid \nu.\, q \xrightarrow{(\nu\,(\varphi),\, \nu\,(\psi))} p''' \mid \nu.\, q'$.

Ax. $\nu.\, \nu.\, [x_0 \leftrightarrow x_1]p \doteq \nu.\, \nu.\, p$. There are the following possible cases.

(1) By rule (Res), $\nu.\, \nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\alpha,\, \nu\,(\varphi))} \nu.\, p'$ and $\nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\delta\,(\alpha),\, \varphi)} p'$, with $\alpha = \bar{x}y, xy, -$. Necessarily, rule (Res) has been applied to $\nu.\, [x_0 \leftrightarrow x_1]p$ and $[x_0 \leftrightarrow x_1]p \xrightarrow{(\delta\,(\delta\,(\alpha)),\, \delta\,(\varphi))} p''$, with $p' = \nu.\, p''$. By rule (Rho), $p \xrightarrow{(\delta\,(\delta\,(\alpha)),\, \varphi')} [x_0 \leftrightarrow x_1]p''$, with $\varphi' = [x_0 \leftrightarrow x_1]\delta\,(\varphi)$.

On the other hand, by rule (Res), $\nu.\, p \xrightarrow{(\delta\,(\alpha),\, \nu\,(\varphi'))} \nu.\, [x_0 \leftrightarrow x_1]p'$ and, again by rule (Res), $\nu.\, \nu.\, p \xrightarrow{(\alpha,\, \nu\,(\nu\,(\varphi')))} \nu.\, \nu.\, [x_0 \leftrightarrow x_1]p''$. The equivalence holds by exploiting axioms E_F and by the fact that $\nu\,(\nu\,(\varphi')) = \nu\,(\varphi)$.

(2) By rule (Open), $\nu.\, \nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\bar{x},\, \delta\,(\varphi))} p'$ and $\nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\bar{x}x_0,\, \delta\,(\varphi))} p'$, with $x \neq x_0$. Necessarily, by rule (Res), $[x_0 \leftrightarrow x_1]p \xrightarrow{(\delta\,(\bar{x})x_1,\, \delta\,(\delta\,(\varphi)))} p''$, with $p' = \nu.\, p''$. By rule (Rho), $p \xrightarrow{(\delta\,(\bar{x})x_0,\, \delta\,(\delta\,(\varphi)))} [x_0 \leftrightarrow x_1]p''$.

On the other hand, by rule (Open), $\nu.\, p \xrightarrow{(\delta\,(\bar{x}),\, \delta\,(\delta\,(\varphi)))} [x_0 \leftrightarrow x_1]p''$ and, by rule (Res$'$), $\nu.\, \nu.\, p \xrightarrow{(\bar{x},\, \delta\,(\varphi))} \nu.\, p''$.

(3) By rule (Res$'$), $\nu.\, \nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\alpha,\, \nu\,(\varphi))} \nu.\, [x_0 \leftrightarrow x_1]p'$ and $\nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\delta\,(\alpha),\, \varphi)} p'$ with $\alpha = \bar{x}, x$.

(a) By rule (Res$'$), $[x_0 \leftrightarrow x_1]p \xrightarrow{(\delta\,(\delta\,(\alpha)),\, \delta\,(\varphi))} p''$, with $p' = \nu.\, [x_0 \leftrightarrow x_1]p''$. By rule (Rho$'$), $p \xrightarrow{(\delta\,(\alpha'),\, \varphi')} p'''$, with $\delta\,(\alpha') = [\delta\,(x_0) \leftrightarrow \delta\,(x_1)]\delta\,(\delta\,(\alpha))$, $\varphi' = [\delta\,(x_0) \leftrightarrow \delta\,(x_1)]\delta\,(\varphi)$, and $p'' = [\delta\,(x_0) \leftrightarrow \delta\,(x_1)]p'''$.

On the other hand, by rule (Res$'$), $\nu.\, p \xrightarrow{(\alpha',\, \nu\,(\varphi'))} \nu.\, [x_0 \leftrightarrow x_1][\delta\,(x_0) \leftrightarrow \delta\,(x_1)]p'''$ and, again by rule (Res$'$), $\nu.\, \nu.\, p \xrightarrow{(\nu\,(\alpha'),\, \nu\,(\nu\,(\varphi')))} \nu.\, [x_0 \leftrightarrow x_1]\nu.\, [x_0 \leftrightarrow x_1][\delta\,(x_0) \leftrightarrow \delta\,(x_1)]p'''$. By axioms E_F it follows that $\nu.\, [x_0 \leftrightarrow x_1]\nu.\, [x_0 \leftrightarrow x_1][\delta\,(x_0) \leftrightarrow \delta\,(x_1)]p''' \doteq \nu.\, [x_0 \leftrightarrow x_1]\nu.\, [x_0 \leftrightarrow x_1]p'''$.

(b) By rule (Open), $\nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\bar{x},\, \delta\,(\varphi))} p'$ and $[x_0 \leftrightarrow x_1]p \xrightarrow{(\bar{x}x_0,\, \delta\,(\varphi))} p'$, with $x \neq x_0$. By rule (Rho), $p \xrightarrow{(yx_1,\, \psi)} p''$, with $[x_1 \mapsto x_0]y = x$, $[x_0 \leftrightarrow x_1]\psi = \varphi$, and $p' = [x_0 \leftrightarrow x_1]p''$.

On the other hand, by rule (Res), $\nu.\, p \xrightarrow{(\nu\,(\bar{y})x_0,\, \nu\,(\psi))} \nu.\, p''$ and, by rule (Open), $\nu.\, \nu.\, p \xrightarrow{(\nu\,(\bar{y}),\, \nu\,(\psi))} \nu.\, p''$.

(4) By rule (Scope), $\nu.\, \nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\nu\,(\varphi),\, \nu\,(\psi))} \nu.\, p'$ and $\nu.\, [x_0 \leftrightarrow x_1]p \xrightarrow{(\varphi,\, \psi)} p'$. Again, by rule (Scope), $[x_0 \leftrightarrow x_1]p \xrightarrow{(\delta\,(\varphi),\, \delta\,(\psi))} p''$, with $p' = \nu.\, p''$. By rule (Rho), $p \xrightarrow{(\varphi',\, \psi')} [x_0 \leftrightarrow x_1]p''$, with $\varphi' = [x_0 \leftrightarrow x_1]\delta\,(\varphi)$ and $\psi' = [x_0 \leftrightarrow x_1]\delta\,(\psi)$.

On the other hand, by applying rule (Scope) twice, $\nu.\, \nu.\, p \xrightarrow{(\nu\,(\nu\,(\varphi')),\, \nu\,(\nu\,(\psi')))} \nu.\, \nu.\, [x_0 \leftrightarrow x_1]p''$. The thesis follows by axioms E_F and by the fact that $\nu\,(\nu\,(\varphi')) = \nu\,(\nu\,([x_0 \leftrightarrow x_1]\delta\,(\varphi))) = \nu\,(\delta\,(\nu\,(\varphi))) = \nu\,(\varphi)$ and similarly for ψ.

Ax. $\rho(\pi.p) \doteq \rho(\pi).\rho p$. Suppose $\pi = \bar{x}y$. By rule (Pre$'$) (similar proof for (Pre)), $\bar{x}y.p \xrightarrow{(\delta\,(z),\, \delta\,(\psi))} (\delta.\,(p \mid \varphi)) \mid x_0 = \delta\,(y)$, with $\psi \sqsubseteq \varphi$ and $\varphi \vdash x = z$. By rule (Rho$'$), $\rho(\bar{x}y.p) \xrightarrow{(\rho_{+1}(\delta\,(z)),\, \rho_{+1}(\delta\,(\psi)))} \rho_{+1}((\delta.\,(p \mid \varphi)) \mid x_0 = \delta\,(y))$.

On the other side, by rule (Pre$'$), $\rho(\bar{x}y).\rho p \xrightarrow{(\delta\,(\rho(z)),\, \delta\,(\rho(\psi)))} \delta.\,(\rho p \mid \rho(\varphi)) \mid x_0 = \delta\,(\rho(y))$. It holds that $\rho_{+1}(\delta\,(z)) = \delta\,(\rho(z))$ and $\rho_{+1}(\delta\,(\psi)) = \delta\,(\rho(\psi))$ and, by axioms E_F, $\rho_{+1}((\delta.\,(p \mid \varphi)) \mid x_0 = \delta\,(y)) \doteq \delta.\,(\rho p \mid \rho(\varphi)) \mid x_0 = \delta\,(\rho(y))$.

If $\pi = F$, the proof is similar but rule (Fus) is applied.

Ax. $\rho(p|q) \doteq \rho p \mid \rho q$. The most interesting case are as follows.

(1) By rule (Rho), $\rho(p|q) \xrightarrow{(\rho(\alpha),\, \rho(\varphi))} \rho p'$ and $p|q \xrightarrow{(\alpha,\, \varphi)} p'$. The most interesting cases are as follows.

(a) By rule (Com′), suppose $p \xrightarrow{(\bar{x}y,\, \varphi)} p''$ and $q \xrightarrow{(\delta\,(x),\, \delta\,(\varphi))} q'$, with $\alpha = \tau$ and $p' = p'' \mid \nu.\,(q' \mid \delta\,(y) = x_0)$.
On the other hand, by rule (Rho) and (Rho′) respectively, $\rho p \xrightarrow{(\rho(\bar{x}y),\, \rho(\varphi))} \rho p''$ and $\rho q \xrightarrow{(\rho_{+1}(\delta\,(x)),\, \rho_{+1}\delta\,(\varphi))} \rho_{+1}q'$.
Then, by rule (Com′), with $\delta\,(\rho(x)) = \rho_{+1}(\delta\,(x))$ and $\delta\,(\rho(\varphi)) = \rho_{+1}(\delta\,(\varphi))$, we obtain $\rho p \mid \rho q \xrightarrow{(\tau,\, \rho(\varphi))} \rho p'' \mid \nu.\,(\rho_{+1}q' \mid \delta\,(\rho(y)) = x_0)$. The thesis follows by applying axioms E_F.

(b) By rule (Close), $p \xrightarrow{(x,\, \psi)} p''$, $q \xrightarrow{(\bar{x},\, \psi)} q'$, with $\alpha = \tau$, $\varphi = \nu\,(\psi)$, and $p' = \nu.\,(p'' \mid q')$.
On the other hand, by rule (Rho′), $\rho p \xrightarrow{(\rho_{+1}(x),\, \rho_{+1}(\psi))} \rho_{+1}p''$ and $\rho q \xrightarrow{(\rho_{+1}(\bar{x}),\, \rho_{+1}(\psi))} \rho_{+1}q'$. Then, by rule (Close), $\rho p \mid \rho q \xrightarrow{(\tau,\, \nu\,(\rho_{+1}(\psi)))} \nu.\,(\rho_{+1}p'' \mid \rho_{+1}q')$. The thesis follows by applying axioms E_F and noting that $\rho(\nu\,(\psi)) = \nu\,(\rho_{+1}(\psi))$.

(2) By rule (Rho′), $\rho(p|q) \xrightarrow{(\rho_{+1}(\overline{x}),\, \rho_{+1}(\varphi))} \rho_{+1}p'$ and $p|q \xrightarrow{(\overline{x},\, \varphi)} p'$. There are the following possible cases.

(a) Suppose that, by rule (Par′), $p \xrightarrow{(\overline{x},\, \varphi)} p''$ (similarly, otherwise) and $p' = p''|\delta.\,q$.
On the other hand, by rule (Rho′), $\rho p \xrightarrow{(\rho_{+1}(\overline{x}),\, \rho_{+1}(\varphi))} \rho_{+1}p''$ and, by rule (Par′), $\rho p \mid \rho q \xrightarrow{(\rho_{+1}(\overline{x}),\, \rho_{+1}(\varphi))} (\rho_{+1}p'') \mid \delta.\,\rho q$.

(b) Suppose that, by rule (Par′$_1$), $p \xrightarrow{(\overline{y},\, \varphi_1)} p''$ and $q \xrightarrow{(-,\, \varphi_2)} q'$ (similarly, otherwise), with $p' = p''|\delta.\,q'$, $\varphi \sqsubseteq \varphi_1 + \delta\,(\varphi_2)$, and $\varphi_1 + \delta\,(\varphi_2) \vdash x = y$.
On the other side, by rule (Rho′) and (Rho), $\rho p \xrightarrow{(\rho_{+1}(\overline{y}),\, \rho_{+1}(\varphi_1))} \rho_{+1}p''$, and $\rho q \xrightarrow{(-,\, \rho(\varphi_2))} \rho q'$, and, by rule (Par′$_1$), $\rho p \mid \rho q \xrightarrow{(\rho_{+1}(\overline{x}),\, \rho_{+1}(\varphi))} \rho_{+1}p'' \mid \delta.\,\rho q'$.

Ax. $\rho\nu.\,p \doteq \nu.\,\rho_{+1}p$. The most interesting cases are as follows.

(1) By rule (Rho), $\rho\nu.\,p \xrightarrow{(\rho(\alpha),\, \rho(\varphi))} \rho p'$ and $\nu.\,p \xrightarrow{(\alpha,\, \varphi)} p'$. Suppose that, by rule (Scope), $\alpha = \nu\,(\psi)$, $\varphi = \nu\,(\varphi')$ and $p \xrightarrow{(\psi,\, \varphi')} p''$ with $p' = \nu.\,p''$.
On the other hand, by rule (Rho), $\rho_{+1}p \xrightarrow{(\rho_{+1}(\psi),\, \rho_{+1}(\varphi))} \rho_{+1}p''$ and, by rule (Scope), $\nu.\,\rho_{+1}p \xrightarrow{(\nu\,(\rho_{+1}(\psi)),\, \nu\,(\rho_{+1}(\varphi)))} \nu.\,\rho_{+1}p''$. The thesis follows by axioms E_F and noting that $\nu\,(\rho_{+1}(\varphi')) = \rho(\nu\,(\varphi')) = \rho(\varphi)$.

(2) By rule (Rho′), $\rho\nu.\,p \xrightarrow{(\rho_{+1}(\alpha),\, \rho_{+1}(\varphi))} \rho_{+1}p'$ and $\nu.\,p \xrightarrow{(\alpha,\, \varphi)} p'$ with $\alpha = \overline{x}, x$. By rule (Res′), $p \xrightarrow{(\delta\,(\alpha),\, \delta\,(\varphi))} p''$ and $p' = \nu.\,[x_0 \leftrightarrow x_1]p''$.
On the other hand, by rule (Rho′), $\rho_{+1}p \xrightarrow{(\rho_{+2}(\delta\,(\alpha)),\, \rho_{+2}(\delta\,(\varphi)))} \rho_{+2}p''$. Then, by rule (Res′), $\nu.\,\rho_{+1}p \xrightarrow{(\rho_{+1}(\alpha),\, \rho_{+1}(\varphi))} \nu.\,[x_0 \leftrightarrow x_1]\rho_{+2}p''$ and, by axioms E_F, $\rho_{+1}\nu.\,[x_0 \leftrightarrow x_1]p'' \doteq \nu.\,[x_0 \leftrightarrow x_1]\rho_{+2}p''$.

Ax. $\rho\, c_{\texttt{rec}\,X.\,P} \doteq c_{\rho(\texttt{rec}\,X.\,P)}$.

By rule (Rec), $[\![P[\texttt{rec}\,X.\,P\,/\,X]]\!] \xrightarrow{(\alpha,\, \varphi)} q$ and $c_{\texttt{rec}\,X.\,P} \xrightarrow{(\alpha,\, \varphi)} q$. Suppose that $\alpha = x.\bar{x}$. By rule (Rho′), $\rho c_{\texttt{rec}\,X.\,P} \xrightarrow{(\rho_{+1}(\alpha),\, \rho_{+1}(\varphi))} \rho_{+1}q$.

On the other hand, $[\![\rho(P[\texttt{rec}\,X.\,P\,/\,X])]\!] \xrightarrow{(\rho_{+1}(\alpha),\, \rho_{+1}(\varphi))} \rho_{+1}q$, and, thus, by rule (Rec), $c_{\rho(\texttt{rec}\,X.\,P)} \xrightarrow{(\rho_{+1}(\alpha),\, \rho_{+1}(\varphi))} \rho_{+1}q$.

Ax. $\delta.\,(\pi.p) \doteq \delta\,(\pi).\delta.\,p$. The proof is similar to the case of $\rho(\pi.p) \doteq \rho(\pi).\rho p$.

Ax. $\delta.\,p \mid q \doteq (\delta.\,p) \mid \delta.\,q$. The proof is similar to the case of $\rho(p|q) \doteq \rho p \mid \rho q$.

Ax. $\delta.\,\nu.\,p \doteq \nu.\,[x_0 \leftrightarrow x_1]\delta.\,p$.

By rule (Del′), $\delta.\,\nu.\,p \xrightarrow{(\delta\,(\alpha),\, \delta\,(\varphi))} [x_0 \leftrightarrow x_1]\delta.\,p'$ and $\nu.\,p \xrightarrow{(\alpha,\, \varphi)} p'$ with $\alpha = \overline{x}, x$. Necessarily, by rule (Res′), $p \xrightarrow{(\delta\,(\alpha),\, \delta\,(\varphi))} p''$, with $p' = \nu.\,[x_0 \leftrightarrow x_1]p''$.

On the other hand, by rule (Del′), $\delta.\,p \xrightarrow{(\delta\,(\delta\,(\alpha)),\, \delta\,(\delta\,(\varphi)))} [x_0 \leftrightarrow x_1]\delta.\,p''$. Then, by rule (Rho′), $[x_0 \leftrightarrow x_1]\delta.\,p \xrightarrow{(\delta\,(\delta\,(\alpha)),\, \delta\,(\delta\,(\varphi)))} [\delta\,(x_0) \leftrightarrow \delta\,(x_1)][x_0 \leftrightarrow x_1]\delta.\,p''$. Finally, by rule (Res′), $\nu.\,[x_0 \leftrightarrow x_1]\delta.\,p \xrightarrow{(\delta\,(\alpha),\, \delta\,(\varphi))} \nu.\,[x_0 \leftrightarrow x_1][\delta\,(x_0) \leftrightarrow \delta\,(x_1)[x_0 \leftrightarrow x_1]\delta.\,p''$. By axioms E_F, $[x_0 \leftrightarrow x_1]\delta.\,\nu.\,[x_0 \leftrightarrow x_1]p'' \doteq \nu.\,[x_0 \leftrightarrow x_1][\delta\,(x_0) \leftrightarrow \delta\,(x_1)][x_0 \leftrightarrow x_1]\delta.\,p''$.

Ax. $\delta.\,\rho p \doteq \rho_{+1}\delta.\,p$. There are two cases.

(1) By rule (Del), $\delta.\,\rho p \xrightarrow{(\delta\,(\alpha),\,\delta\,(\varphi))} \delta.\,p'$, with $\delta\,(\alpha) \neq \bar{x}, x$. Then, $\rho p \xrightarrow{(\rho(\alpha'),\,\rho(\varphi'))} \rho p''$ and $p \xrightarrow{(\alpha',\,\varphi')} p''$, with $\rho p'' = p'$, $\alpha = \rho(\alpha')$, $\varphi = \rho(\varphi')$, and $\alpha' \neq \bar{x}', x'$, for any x'.

On the other hand, by rule (Del), $\delta.\,p \xrightarrow{(\delta\,(\alpha'),\,\delta\,(\varphi'))} \delta.\,p''$ with $\alpha' \neq \bar{x}', x'$ and, by rule (Rho), $\rho_{+1}\delta.\,p \xrightarrow{(\rho_{+1}(\delta\,(\alpha'),\,\rho_{+1}(\delta\,(\varphi'))} \rho_{+1}\delta.\,p''$.

(2) By rule (Del′), suppose $\delta.\,\rho p \xrightarrow{(\delta\,(\bar{x}),\,\delta\,(\varphi))} [x_0 \leftrightarrow x_1]\delta.\,p'$ (similarly, if $\alpha = \delta\,(x)$). Necessarily, $\rho p \xrightarrow{(\rho(\bar{x}'),\,\rho(\varphi))} \rho_{+1}p''$ and $p \xrightarrow{(\bar{x}',\,\varphi')} p''$, with $\rho(\bar{x}') = \bar{x}$, $\rho(\varphi') = \varphi$ and $p' = \rho_{+1}p''$.

On the other hand, by rule (Del′), $\delta.\,p \xrightarrow{(\delta\,(\bar{x}'),\,\delta\,(\varphi'))} [x_0 \leftrightarrow x_1]\delta.\,p''$ and $\rho_{+1}\delta.\,p \xrightarrow{(\rho_{+1}(\delta\,(\bar{x}')),\,\rho_{+1}(\delta\,(\varphi')))} \rho_{+2}[x_0 \leftrightarrow x_1]\delta.\,p''$. Note that $\rho_{+2}[x_0 \leftrightarrow x_1]\delta.\,p'' = [x_0 \leftrightarrow x_1]\rho_{+2}\delta.\,p''$, as ρ_{+2} does not substitute either x_0 or x_1.

Ax. $\delta.\,c_{\mathtt{rec}}\,X.\,P \doteq c_{\delta\,(\mathtt{rec}\,X.\,P)}$. The proof is similar to the case of $\rho\, c_{\mathtt{rec}\,X.\,P} \doteq c_{\rho(\mathtt{rec}\,X.\,P)}$.

The proof of the theorem is trivial in the remaining cases, that is, for axioms in **(group)**, **(par)**, **(fus)** and for those ones involving term **0**.

B. Proof of Theorem 3.16

We now give a formal proof of Theorem 3.16, by detailing the steps that we have outlined in Section 3.

Definition B.1. Let σ be a substitution. The syntactical application of σ to any term p of algebra B_F 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 $\mathtt{Eq}(p)$. For example, let $p = (x = y)\,|\,(y = z)\,|\,\bar{x}w.\mathbf{0}$. For $\sigma = [y \mapsto z, x \mapsto z]$ a substitutive effect of $\mathtt{Eq}(p)$, $\sigma(p) \doteq \bar{z}w.\mathbf{0}$.

Definition B.2. The transition system lts_1 is defined as $lts_1 = \langle B_F, \longrightarrow_1 \rangle$, where $\longrightarrow_1$ is given by the rules in Table B.1.

The first group of rules in Table B.1 is similar to those given in Table 2. The aim of lts_1 is to ensure saturation of behaviours of a term only with respect to the explicit fusions contained in the term. Thus, rules for prefixes do not consider all the possible fusion contexts. Moreover, lts_1 contains rule (Eq) in place of the rules for combination and propagation of explicit fusions in Table 3. In fact, it can be easily seen that rule (Eq) has the same effect of the rules in Table 3, but it is not in De Simone format.

Definition B.3 (*bisimilarity* $\sim_1$). A *bisimulation on* lts_1 is a binary symmetric relation $\mathcal{S}$ between terms of B_F 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. For any fusion agent P, neither $[\![P]\!]$ nor p', for all p' such that $[\![P]\!] \xrightarrow{\alpha}_1 p'$, contain explicit fusions under prefixes. Thus, for our purpose, we can restrict to the algebra B_F^*, that is obtained from B_F by removing terms containing explicit fusions under prefixes.

Lemma B.4. *Let p be a term of algebra B_F^* and let σ be a substitutive effect of* $\mathtt{Eq}(p)$. *Then, $p \xrightarrow{\alpha}_1 p'$ with $\alpha \neq \bar{x}, x$ (resp. $\alpha = \bar{x}, x$) if and only if $\sigma(p) \xrightarrow{\sigma(\alpha)}_1 \sigma(p')$ (resp. $\sigma(p) \xrightarrow{\sigma_{+1}(\alpha_{+1})}_1 \sigma_{+1}(p')$).*

Table B.1
Transition system lts_1

(Rho) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \alpha \neq x, \bar{x}}{\rho p \xrightarrow{\rho(\alpha)}_1 \rho q}$	(Rho′) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \alpha = \bar{x}, x}{\rho p \xrightarrow{\rho_{+1}(\alpha)}_1 \rho_{+1} q}$
(Del) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \alpha \neq x, \bar{x}}{\delta .\, p \xrightarrow{\delta(\alpha)}_1 \delta .\, q}$	(Del′) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \alpha = \bar{x}, x}{\delta .\, p \xrightarrow{\delta(\alpha)}_1 [x_0 \leftrightarrow x_1]\delta .\, q}$
(Par) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \alpha \neq x, \bar{x}, -}{p \mid r \xrightarrow{\alpha}_1 q \mid r}$	(Par′) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \alpha = x, \bar{x}}{p \mid r \xrightarrow{\alpha}_1 q \mid \delta .\, r}$
(Res) $\dfrac{p \xrightarrow{\delta(\alpha)}_1 q \quad \alpha = \bar{x}y, xy, -}{\nu .\, p \xrightarrow{\alpha}_1 \nu .\, q}$	(Res′) $\dfrac{p \xrightarrow{\delta(\alpha)}_1 q \quad \alpha = \bar{x}, x}{\nu .\, p \xrightarrow{\alpha}_1 \nu .\, [x_0 \leftrightarrow x_1] q}$
(Open) $\dfrac{p \xrightarrow{x x_0}_1 q \quad x \neq x_0}{\nu .\, p \xrightarrow{x}_1 q}$	(Scope) $\dfrac{p \xrightarrow{\varphi}_1 q}{\nu .\, p \xrightarrow{\nu(\varphi)}_1 \nu .\, q}$
(Rec) $\dfrac{[\![P[\mathtt{rec}\, X.\, P \,/\, X]]\!] \xrightarrow{\alpha}_1 q}{c_{\mathtt{rec}\, X.\, P} \xrightarrow{\alpha}_1 q}$	
(Com) $\dfrac{p_1 \xrightarrow{xy}_1 q_1 \quad p_2 \xrightarrow{\bar{x}'z}_1 q_2}{p_1 \mid p_2 \xrightarrow{y=z}_1 q_1 \mid q_2 \mid y = z} \quad \mathtt{Eq}(p_1 \mid p_2) \vdash x = x'$	
(Com′) $\dfrac{p_1 \xrightarrow{xy}_1 q_1 \quad p_2 \xrightarrow{\delta(\bar{x}')}_1 q_2}{p_1 \mid p_2 \xrightarrow{\tau}_1 q_1 \mid \nu .\, (q_2 \mid \delta(y) = x_0)} \quad \mathtt{Eq}(p_1 \mid p_2) \vdash x = x'$	
(Close) $\dfrac{p_1 \xrightarrow{x}_1 q_1 \quad p_2 \xrightarrow{\bar{x}'}_1 q_2}{p_1 \mid p_2 \xrightarrow{\tau}_1 \nu .\, (q_1 \mid q_2)} \quad \mathtt{Eq}(p_1 \mid p_2) \vdash x = x'$	
(Pre) $xy.p \xrightarrow{xy}_1 p$	(Pre′) $xy.p \xrightarrow{\delta(x)}_1 \delta .\, p \mid \delta(y) = x_0$
(Fus) $\varphi.p \xrightarrow{\varphi}_1 p \mid \varphi$	(Eq) $\dfrac{p \xrightarrow{\alpha}_1 q \quad \mathtt{Eq}(p) \vdash \alpha = \beta}{p \xrightarrow{\beta}_1 q}$

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

Proof (*Hint*). By induction on the rules of lts_1. □

For instance, consider $p = (x = y) \mid \bar{z}v.\mathbf{0}$ and $q = (z = y) \mid xw.\mathbf{0}$. Since $\mathtt{Eq}(p \mid q) \vdash z = x$, by rule (Com) $p \mid q \xrightarrow{\{v=w\}}_1 x = y \mid z = y \mid v = w$. On the other hand, any substitutive effect σ of $\mathtt{Eq}(p \mid q)$ fuses z and x and, thus, for example, for $\sigma = [z \mapsto x]$, $\sigma(p \mid q) \xrightarrow{\{v=w\}}_1 x = y \mid x = y \mid v = w$.

Lemma B.5. *Let p be a term of algebra B_F^* and let σ be a substitutive effect of $\mathtt{Eq}(p)$. If $p \xrightarrow{\alpha}_1 q$ and $\alpha = \bar{x}y$, xy then $\mathtt{Eq}(q) = \mathtt{Eq}(p)$; if $p \xrightarrow{\alpha}_1 q$ and $\alpha = \bar{x}, x$ then $\mathtt{Eq}(q) = \delta(\mathtt{Eq}(p))$; if $p \xrightarrow{\varphi}_1 q$ then $\mathtt{Eq}(q) = \mathtt{Eq}(p) + \varphi$.*

Proof (*Hint*). By induction on the rules of lts_1. □

Lemma B.6. *Let p and q be terms of algebra B_F^* 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 (*Hint*). It follows by Lemmas B.4 and B.5. □

Lemma B.7. *Let P and Q be two fusion agents. If $P \dot\sim Q$, then $\delta(P) \dot\sim \delta(Q)$ and $\rho(P) \dot\sim \rho(Q)$.*

Lemma B.8

(1) *Let P and Q be fusion agents. If $P \overset{\gamma}{\longmapsto} Q$ and $\gamma = \overline{x}\, y, x\, y$ then $[\![P]\!] \overset{\gamma}{\longrightarrow}_1 [\![Q]\!]$; if $P \overset{\bar{x}(y)}{\longmapsto} Q$ (resp. $P \overset{x(y)}{\longmapsto} Q$) then $[\![P]\!] \overset{\delta(\bar{x})}{\longrightarrow}_1 [\delta(y) \leftrightarrow x_0]\delta.\, [\![Q]\!]$ (resp. $[\![P]\!] \overset{\delta(x)}{\longrightarrow}_1 [\delta(y) \leftrightarrow x_0]\delta.\, [\![Q]\!]$); if $P \overset{\varphi}{\longmapsto} Q$, then $[\![P]\!] \overset{\varphi}{\longrightarrow}_1 [\![Q]\!] \mid \varphi$.*

(2) *Let p and q be in B_F^* and let σ be a substitutive effect of $\mathtt{Eq}(p)$. If $p \overset{\alpha}{\longrightarrow}_1 q$ and $\alpha = \bar{x}y, xy$, then $\{\sigma(p)\} \overset{\sigma(\alpha)}{\longmapsto} \{\sigma(q)\}$; if $p \overset{\overline{x}}{\longrightarrow}_1 q$ (resp. $p \overset{x}{\longrightarrow}_1 q$), then $\{\sigma(p)\} \overset{\sigma(\nu(\overline{x}))x_i}{\longmapsto} \{\sigma(\nu([\delta(x_i) \leftrightarrow x_0](q)))\}$, (resp. $\{\sigma(p)\} \overset{\sigma(\nu(x))x_i}{\longmapsto} \{\sigma(\nu([\delta(x_i) \leftrightarrow x_0](q)))\}$, for every x_i such that $\delta(x_i) \notin \mathrm{fn}(\{p\})$; if $p \overset{\varphi}{\longrightarrow}_1 q \mid \varphi$ then $\{\sigma(p)\} \overset{\sigma(\varphi)}{\longmapsto} \{\sigma(q)\}$.*

Proof (*Hint*). By induction on the rules of the fusion transition system and on the rule of lts_1, respectively. □

Theorem B.9. *Let p and q be terms of algebra B_F^* 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 $[\![\cdot]\!]$.*

Proof. It follows by Lemmas B.6, B.7, and B.8. □

Corollary B.10. *Let P and Q be two fusion agents. $P \dot\sim Q$ if and only if $[\![P]\!] \sim_1 [\![Q]\!]$.*

Definition B.11. The transition system lts_2 is defined as $lts_2 = \langle B_F, \longrightarrow_2 \rangle$, where $\longrightarrow_2$ is given by adding to the rules of lts_1 a rule for closing with respect to fusions in parallel:

$$\text{(Ctx)}\ \frac{p \mid \varphi \overset{\alpha}{\longrightarrow}_1 q}{p \overset{\alpha,\varphi}{\longrightarrow}_2 q}$$

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

Lemma B.12. *Let p and q be two terms of algebra B_F. If $p \sim_2 q$ then $p \mid \varphi \sim_2 q \mid \varphi$, for all φ.*

Proof. By induction on the rules of lts_2. □

Theorem B.13. *$P \sim_{he} Q$ if and only if $[\![P]\!] \sim_2 [\![Q]\!]$, where $\sim_{he}$ denotes fusion hyperequivalence.*

Proof. The proof follows by Corollary B.10 and by Lemma B.12. □

Definition B.14. The third transition system lts_3 is defined as $lts_3 = \langle B_F, \longrightarrow_3 \rangle$, where $\longrightarrow_3$ is given by the rules in Table 2 and 4 plus rule

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

Bisimilarity $\sim_3$ is analogous to $\sim_2$ ($\longrightarrow_2$ replaces $\longrightarrow_3$).

The proof of the Theorem 3.16 is concluded by showing that $\sim_3$ is equivalent to both $\sim_2$ and $\sim_F$.

Lemma B.15. *Let p and q be two terms of algebra B. If $p \sim_3 q$ then $p \mid \varphi \sim_3 q \mid \varphi$, for all φ.*

Proof. By induction on the rules of lts_3. □

Theorem B.16. *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 Lemmas B.12 and B.15. □

Theorem B.17. *Let p and q be two terms of algebra B. $p \sim_3 q$ if and only if $p \sim_F q$.*

Proof. The intuition behind the proof is that the rules of lts_F in Table 3 can be simulated by rule (Eq) in lts_3 and viceversa. Moreover, lts_F ensures that if $p \sim_F q$ then $\mathtt{Eq}(p) = \mathtt{Eq}(q)$. □

References

[1] L. Aceto, W.J. Fokkink, C. Verhoef, Structural operational semantics, in: Handbook of Process Algebra, Elsevier, 2002.
[2] M. Boreale, M.G. Buscemi, U. Montanari, D-Fusion: a distinctive fusion calculus, in: Proc. of APLAS'04, LNCS, vol. 3302, Springer, 2004.
[3] M. Boreale, M.G. Buscemi, U. Montanari, A general name binding mechanism, in: Proc. of TGC '05, LNCS, vol. 3705, Springer, 2005.
[4] M.G. Buscemi, U. Montanari, A first order coalgebraic model of pi-calculus early observational equivalence, in: Proc. of CONCUR '02, LNCS, vol. 2421, Springer, 2002 (Extended version available as TR-02-14, Dipartimento di Informatica, University of Pisa).
[5] M.G. Buscemi, U. Montanari, A compositional coalgebraic model of a fragment of fusion calculus, in: ENTCS, Elsevier, in press.
[6] M.G. Buscemi, U. Montanari, CC-Pi: a constraint-based language for specifying service level agreements, Manuscript.
[7] A. Corradini, M. Große-Rhode, R. Heckel, Structured transition systems as lax coalgebras, in: Proc. of CMCS'98, ENTCS, vol. 11, Elsevier Science, 1998.
[8] A. Corradini, R. Heckel, U. Montanari, Compositional SOS and beyond: a coalgebraic view of open systems, Theor. Comput. Sci. 280 (2002) 163–192.
[9] M.P. Fiore, S. Staton, A congruence rule format for name-passing process calculi from mathematical operational semantics, in: Proc. of LICS'06, IEEE Computer Society Press, 2006.
[10] M.P. Fiore, D. Turi, Semantics of name and value passing, in: Proc. of LICS'01, IEEE Computer Society Press, 2001.
[11] F. Gadducci, M. Miculan, U. Montanari, Some characterization results for permutation algebras, in: Proc. of Workshop COMETA, ENTCS, vol. 104, Elsevier, 2004.
[12] P. Gardner, C. Laneve, L. Wischik, The fusion machine (extended abstract), in: Proc. of CONCUR '02, LNCS, vol. 2421, Springer-Verlag, 2002.
[13] N. Ghani, K. Yemane, B. Victor, Relationally staged computation in calculi of mobile processes, in: Proc. of CMCS'04, ENTCS, vol. 106, Elsevier, 2004.
[14] M. Miculan, K. Yemane, A unifying model of variables and names, in: Proc. of FOSSACS'05, LNCS, vol. 3441, Springer, 2005.
[15] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes (parts I and II), Inform. Comput. 100 (1) (1992) 1–77.
[16] U. Montanari, M. Pistore, Pi-calculus, structured coalgebras and minimal HD-automata, in: Proc. of MFCS'00, LNCS, vol. 1983, Springer, 2000.
[17] U. Montanari, M. Pistore, Structured coalgebras and minimal HD-automata for the pi-calculus, Theor. Comput. Sci. 340 (3) (2005) 539–576.
[18] M.R. Mousavi, M.A. Reniers, Congruence for structural congruences, in: Proc. of FOSSACS'05, LNCS, vol. 3441, Springer, 2005.
[19] J. Parrow, B. Victor, The fusion calculus: expressiveness and symmetry in mobile processes, in: Proc. of LICS'98, IEEE Computer Society Press, 1998.
[20] G.D. Plotkin, A structural approach to operational semantics, J. Logic Algebr. Program. 60–61 (2004) 17–139.
[21] J.J.M.M. Rutten, Universal coalgebra: a theory of systems, Theor. Comput. Sci. 249 (1) (2000) 3–80.
[22] D. Sangiorgi, A theory of bisimulation for the pi-calculus, Acta Inform. 33 (1) (1996) 69–97.
[23] D. Turi, G. Plotkin, Towards a mathematical operational semantics, in: Proc. of LICS'97, IEEE Computer Society Press, 1997.
[24] B. Victor, The fusion calculus: expressiveness and symmetry in mobile processes, Ph.D. thesis, Department of Computer Systems, Uppsala University, 1998.
[25] L. Wischik, P. Gardner, Explicit fusions, Theor. Comput. Sci. 340 (3) (2005) 606–630.
[26] L. Wischik, P. Gardner, Strong bisimulation for the explicit fusion calculus, in: Proc. of FoSSaCS'04, LNCS, vol. 2987, Springer, 2004.