Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 162 (2006) 135–139
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

A Compositional Coalgebraic Model of a Fragment of Fusion Calculus[1]

Maria Grazia Buscemi[2] and Ugo Montanari[3]

Dipartimento di Informatica, University of Pisa, Italy

Abstract

This work is a further step in exploring the labelled transitions and bisimulations of fusion calculi. We follow the approach developed by Turi and Plotkin for lifting transition systems with a syntactic structure 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 to give an account of explicit fusions through labelled transitions. In this short essay, we focus on a fragment of the fusion calculus without recursion and replication.

Keywords: Process Calculi, Algebraic/Coalgebraic Models

1 Introduction

The fusion calculus [6] has been introduced as a variant of the pi-calculus [4] that makes input and output operations fully symmetric and that enables a more general name matching mechanism during synchronisation. A fusion is a name equivalence that allows to use interchangeably in a term all names of 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 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 it has the effect of a (possibly non-injective) name substitution. On the other hand, the explicit fusion calculus [3] is a variant that aims at guaranteeing asynchronous broadcasting of fusions to the environment. Explicit

[1] Research partially supported by the EU IST-FP6 16004 Integrated Project *SENSORIA*.
[2] Email: buscemi@di.unipi.it
[3] Email: ugo@di.unipi.it

doi:10.1016/j.entcs.2005.12.109

fusions are processes that exist concurrently with the rest of the system and enable to freely use two names one for the other.

A coalgebraic framework [7] presents several advantages: morphisms between coalgebras (cohomomorphisms) enjoy the property of "reflecting behaviours" and thus they allow, for example, to characterise bisimulation equivalences as kernels of morphisms and bisimilarity as the kernel of the morphism to the final coalgebra. Also adequate temporal logics and proof methods by coinduction fit nicely into the picture.

However, in the ordinary coalgebraic framework, the states of transition systems are seen simply as set elements, i.e. the algebraic structure needed for composing programs and states is disregarded. Bialgebraic models take a step forward in this direction: they aim at capturing interactive systems which are compositional. Roughly, bialgebras [8] 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. Turi and Plotkin in [8] have proved that a transition system *lts* with a syntactic structure can be lifted to a bialgebra, provided that the SOS rules of *lts* are in GSOS rule format. As a consequence, bisimilarity on *lts* is a congruence, namely, compositionality of abstract semantics is automatically preserved.

We apply the general approach developed in [8] to provide a compositional coalgebraic model of a fragment of the fusion calculus without recursion and restriction. 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 our result does not only concern the fusion calculus but it could fit within theoretical foundations of languages based on pattern matching.

We focus on a fragment of the fusion calculus since, for the purpose of this short essay, we are only interested in addressing the key issues of name fusions. The introduction of restriction requires handling dynamic creation of names, that is an orthogonal aspect to name fusions and has been considered in [1] for the pi-calculus. In any case, restriction and recursion can be modelled within our theory. We refer to [2] for the coalgebraic model of the full fusion calculus.

We first introduce an algebra whose operations are the constructs of the calculus plus constants modelling explicit fusions. We then define a transition system equipped with that syntactic structure and conclude that the associated bisimilarity is a congruence. Remarkably enough, explicit fusions enable us to model global effects of name fusions in the fusion calculus, even if our algebra does not contain substitution operations. Indeed, observable effects of substitutions are simulated by special SOS rules which saturate process behaviours, while still keeping the nice property of asynchronous propagation typical of explicit fusions. We claim that the translation of fusion agents in our algebra is fully abstract with respect to fusion hyperequivalence. For lack of space, we omit proofs; they can be found in [2].

2 A Labelled Transition System for Fusion Calculus

The fusion calculus is a variant of the pi-calculus. 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. For example, the interaction between two 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:

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

In this work, we consider a monadic version of the calculus without restriction and replication. For a full treatment of the fusion calculus we refer to [6].

Let $\mathfrak{N} = \{x_0, x_1, x_2, \ldots\}$ be the infinite, countable, totally ordered set of *names* and let $x, y, z \ldots$ denote names. A *fusion* is a total equivalence relation on $\mathfrak{N}$ with only finitely many non-singular equivalence classes. Fusions are ranged over by $\varphi, \psi, \ldots$ and τ denotes the identity fusion. By $\varphi + \psi$ we denote the finest fusion which is coarser than φ and ψ, that is the reflexive and transitive closure $(\varphi \cup \psi)^\star$; by $\varphi[x]$ we denote the equivalence class of x in φ; by $\varphi \sqsubseteq \psi$ we mean that φ is finer that ψ, that is, $\varphi[x] \subseteq \psi[x]$ for all $x \in \mathfrak{N}$; we write $\{x = y\}$ to denote $\{(x, y), (y, x)\}$.

Definition 2.1 Let A be the initial algebra T_Σ with $\Sigma ::= \mathbf{0} \mid \pi.\_ \mid \_|\_ \mid x = y$, where prefixes are defined as $\pi ::= \bar{x}y \mid xy \mid \varphi$.

Note that, even if the algebra does not contain substitution operations, explicit fusions $x = y$ in the signature allow to model substitutive effects of fusion calculus. 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 with y or viceversa in the processes in parallel, that is names x and y can be used one for the other in the context $x = y \,|\, \_$. In practice, rather than applying to an agent the substitutive effect of a fusion, the agent is run in parallel with the fusion itself. Fusion agents can be translated into terms of algebra A as expected.

Definition 2.2 We let L be the set of labels $L = \Lambda \times \Phi$, where $\Lambda = \{x\,y,\ \overline{x}\,y,\ \varphi,\ - \mid x, y, \mathrm{n}(\varphi) \in \mathfrak{N}\}$ and Φ is the set of all fusions over $\mathfrak{N}$. We let $\alpha, \beta, \ldots$ range over Λ.

The left-hand components of the labels L correspond to the free actions of the fusion calculus, while the right-hand components φ are introduced to express that two names in the same equivalence class of φ can be used interchangeably in a given term.

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

Definition 2.3 [transition specification Δ] The transition specification Δ is the tuple $\langle \Sigma, L, R \rangle$, where the signature Σ is as in Definition 2.1, labels L are as in Definition 2.2 and R is the set of SOS rules in Table 1. Transitions take the form $p \xrightarrow{(\alpha,\varphi)} q$, where (α, φ) ranges over L.

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

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

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

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

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

$$(\textsc{Par})\quad \frac{p \xrightarrow{(\alpha,\varphi)} q}{p \,|\, r \xrightarrow{(\alpha,\varphi)} q \,|\, r} \qquad\qquad (\textsc{Com})\quad \frac{p_1 \xrightarrow{(xy,\varphi)} q_1 \quad p_2 \xrightarrow{(\bar{x}z,\varphi)} q_2}{p_1 \,|\, p_2 \xrightarrow{(y=z,\varphi)} q_1 \,|\, q_2 \,|\, y = z}$$

Rule (Pre) is analogous with output actions.

Table 1
Structural Operational Semantics

The crucial rules in Table 1 are those ones for dealing with explicit fusions. By rule (Exp) explicit fusions are propagated and by rules ($\textsc{Par}_f$) and ($\textsc{Par}_1$) they are combined with each other and with other agents in parallel, respectively. Rules (Pre) and (Fus) are intended to ensure that the associated bisimilarity be preserved by closure with respect to fusions running in parallel. All side conditions ensure a saturation of process behaviours with respect to the explicit fusions.

Example 2.4

- Let p be the term $p = \overline{x}\,y.y\,w.\mathbf{0}$. By rule (Pre), p can undergo any of the following transitions:

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

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

Theorem 2.5 *Let lts be the transition system lts $= \langle A, \longrightarrow \rangle$, where $\longrightarrow$ is defined by the SOS rules in Table 1, and let $\sim$ be the bisimilarity associated on lts.*

Bisimilarity $\sim$ is a congruence.

Theorem 2.6 *Let P and Q be two fusion agents. Then, $P \sim_{he} Q$ iff $[\![P]\!] \sim [\![Q]\!]$, where $\sim_{he}$ denotes fusion hyperequivalence [6] and $[\![\cdot]\!]$ is the translation of fusion agents into terms of A.*

3 Conclusions

For the purpose of this paper we have considered a fragment of the fusion calculus. In [2] we propose a bialgebraic model of the full calculus, which makes a more complex scenario. The restriction operation, for instance, introduces issues of dynamic name creation. For this reason, in *loc.cit.*, the authors define a permutation algebra [5,1] enriched with the operations of the calculus and explicit fusions, and equipped with an axiomatisation. In this more general case, bisimilarity is proved to be a congruence, by exploiting a lifting result [1] that generalises the approach by Turi and Plotkin to calculi with structural axioms.

References

[1] M. Buscemi and U. Montanari. A First Order Coalgebraic Model of Pi-Calculus Early Observational Equivalence. In *Proc. of CONCUR '02*, LNCS 2421. Springer, 2002. Extended version available as TR-02-14, Dipartimento di Informatica - University of Pisa.

[2] M. Buscemi and U. Montanari. A Compositional Coalgebraic Model of Fusion Calculus. Manuscript. Available at `http://www.di.unipi.it/~buscemi/Fusion.pdf`. To appear in *Journal of Logic and Algebraic Programming*.

[3] P. Gardner and L. Wischik. Explicit Fusions. *Theoretical Computer Science*, 340(3):606–630, 2005.

[4] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (parts I and II). *Information and Computation*, 100(1):1–77, 1992.

[5] U. Montanari and M. Pistore. Structured Coalgebras and Minimal HD-Automata for the pi-Calculus. *Theoretical Computer Science*, 340(3): 539-576, 2005.

[6] J. Parrow and B. Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. In *Proc. of LICS'98*. IEEE Computer Society Press, 1998.

[7] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science* 249(1):3–80, 2000.

[8] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. of LICS'97*, IEEE. Computer Society Press, 1997.