

G-Reactive Systems as Coalgebras

Filippo Bonchi, Ugo Montanari ¹

*Department of Computer Science
University of Pisa*

Abstract

The semantics of process calculi has traditionally been specified by labelled transition systems (LTSs), but with the development of name calculi it turned out that definitions employing reduction semantics are sometimes more natural. Reactive Systems à la Leifer and Milner allow to derive from a reduction semantics definition an LTS equipped with a bisimilarity relation which is a congruence. This theory has been extended to G-Reactive Systems by Sassone and Sobocinski in order to properly handle structural equivalence.

Universal Coalgebra provides a categorical framework where bisimilarity can be characterized as final semantics, i.e., each LTS can be mapped to a minimal realization identifying bisimilar states. Moreover, it is often possible to lift coalgebras to an algebraic setting (yielding bialgebras by Turi and Plotkin or, slightly more generally, structured coalgebras by Corradini, Heckel and Montanari) with the property that bisimilarity is compositional with respect to the lifted structure. The existence of minimal realizations is of theoretical interest, but it is even more of practical interest whenever LTSs are employed for finite state verification.

In this paper we show that for every G-Reactive System we can build a coalgebra. Furthermore, if bisimilarity is compositional in the Reactive System, then we can lift this coalgebra to a structured coalgebra.

Keywords: Process calculus, labelled transition system, reactive systems, G-reactive systems, universal coalgebra

1 Introduction

The operational semantics of process calculi is usually given in terms of transition systems labeled with actions, which, when visible, represent both observations and interactions with the external world. The abstract semantics is given in terms of behavioral equivalences, which depend on the action labels and on the amount of branching structure considered. Behavioral equivalences are often congruences with respect to the operations of the language, and this property, which depends on how actions are combined and transformed by the operations, expresses the compositionality of the abstract semantics.

A simpler approach, inspired by classical formalisms like λ -calculus, Petri nets, term and graph rewriting, and pioneered by the Chemical Abstract Machine [4],

¹ Research partially supported by the IST 2004-16004 SEnSOria, and the MIUR PRIN 2005015824 ART.

defines operational semantics by means of *structural axioms* and *reduction rules*. Process calculi representing complex systems, in particular those able to generate and communicate names, are often defined in this way, since structural axioms give a clear idea of the intended structure of the states while reaction rules, which are often non conditional, give a direct account of the possible steps. Transitions caused by reaction rules, however, are not labeled, since they represent evolutions of the system without interactions with the external world. Thus reduction semantics in itself is neither abstract nor compositional.

To enhance the expressiveness of reduction semantics, Leifer and Milner proposed in [14] the *theory of Reactive Systems*: a systematic method for deriving a labeled transition system from reduction rules. The main idea is the following: a process p can do a move with label $C[-]$ and become p' iff $C[p] \rightsquigarrow p'$. Roughly a Reactive System is a syntactical category equipped with a set of rewriting rules. In the category, arrows represents terms and contexts while arrows composition models the insertion of terms into contexts. Rewriting rules are pairs of terms, and the reduction relation \rightsquigarrow is defined by closing the rules under contexts composition.

Leifer and Milner introduced also the categorical notions of relative pushout (RPO) and idem relative pushout (IPO) in order to specify a/the minimal context that allows the state to react with a given rule. This construction leads to labelled transition systems (LTS) that use only contexts generated by IPOs, as labels. Bisimilarity, as well as trace and failure equivalence, on this LTS is a congruence whenever the syntactical category underlying the Reactive Systems has RPOs.

Sassone and Sobociński have pointed out that usually syntactical categories quotiented w.r.t. structural axioms have not RPOs (Example 2.2.2 in [19]). Moreover considering terms up to structural equivalence usually makes to derive wrong labeled transition system (Example 2.2.1 in [19]). Thus they extend the theory to G-Reactive Systems that are standard Reactive Systems, but defined on G-categories, i.e., categories where arrows are terms not quotiented w.r.t. structural axioms, and between structurally equivalent arrows there are 2-cells, that represent proofs of structural equivalence.

The concepts of RPO and IPO, previously defined by Leifer and Milner, corresponds in G-category to groupoidal relative pushout (GRPO) and to groupoidal idem pushout (GIPO). Instead of generating the LTS by IPOs, Sassone and Sobociński use GIPOs and prove that bisimilarity is a congruence whenever the base G-category has GRPOs. In [18], the same authors have shown that the theory of G-Reactive Systems exactly subsumes Borrowed Contexts Rewriting [10].

The aim of this paper is to recast G-Reactive Systems as *structured coalgebras*. The use of coalgebras for the specification of dynamical systems with a hidden state space is receiving more and more attention in the last years, as a valid alternative to algebraic methods based on observational equivalences [15]. Given an endofunctor \mathbf{F} on a category \mathbf{C} , a coalgebra is an arrow $f: X \rightarrow \mathbf{F}(X)$ of \mathbf{C} and a coalgebra morphism from f to f' is an arrow $h: X \rightarrow X'$ of \mathbf{C} with $h; f' = f; \mathbf{F}(h)$. Under certain conditions on \mathbf{C} and \mathbf{F} , a category of coalgebras admits a final object, which can be considered informally as the minimal realization of the union of all

the coalgebras in the category.

Ordinary labeled transition systems (with finite or countable branching) can be represented as coalgebras for a suitable functor on **Set**. Furthermore, the unique morphism to the final coalgebra induces an equivalence which turns out to be exactly bisimilarity. Thus a first (rather straightforward) result of this paper is to show that the labeled transition systems derived from Reactive Systems can be considered as coalgebras and that their bisimilarity can be characterized as final semantics.

However, this representation forgets about the algebraic structure of Reactive Systems, i.e., the possibility of contextualizing a term. As a consequence, the property that bisimilarity is a congruence, which is essential for making abstract semantics compositional, is not reflected in the structure of the model.

The problem of integrating coalgebras and algebras obtaining a model equipped with both structures has been tackled in [20], and an alternative but equivalent approach based on *structured coalgebras* is presented in [7,8]. Here, the endofunctor determining the coalgebraic structure is lifted from **Set** to the category of Γ -algebras, for some algebraic theory Γ . Morphisms between coalgebras in this category are both Γ -homomorphisms and coalgebra morphisms, and thus the unique morphism to the final coalgebra, which always exists, induces a (coarsest) bisimulation congruence on any coalgebra.

It turns out that the conditions that guarantee the compositionality of bisimilarity in the theory of G-Reactive Systems (i.e., the existence of GIPOs) imply the existence of a structured coalgebras equivalent to the distilled LTS. Thus a second result of the paper is to provide a different understanding of why bisimilarity is a congruence in the derived LTS. Namely the derived transition system is functorial, i.e., it preserves arrows composition. Here the *decomposition property* of GIPO's is pivotal and it reminds us the decomposition property of tile systems [11] that guarantees compositionality of tile bisimilarity.

In [6], we have provided a coalgebraic characterization of standard Reactive Systems *à la* Leifer and Milner. Thus the present paper has to be understood as an extension of [6] to G-Reactive Systems. We will show that the extension is straightforward: instead of using the decomposition property of IPOs, we use the decomposition property of GIPOs.

The paper is structured as follows. In Section 2, we introduce some basic notions on 2-categories and G-categories that are fundamental in order to introduce G-Reactive Systems in Section 3. In Section 4 we introduce coalgebras and structured coalgebras. Section 5 defines a coalgebra for the LTS derived from the Reactive System and Section 6 proves that, if the base category has GRPOs, then the coalgebra previously defined is a structured coalgebra, i.e., bisimilarity is a congruence.

2 Basic facts on 2-categories

In this section we introduce the basic concepts of 2-categories and G-categories. This will be fundamental in order to introduce the theory of G-Reactive Systems in the next section. For a more detailed introduction on the topic, we refer the reader

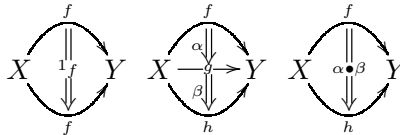
to [13].

In the following we use $\mathbf{C}[X, Y]$ to denote the *homset* X, Y , i.e., the collection of arrows of \mathbf{C} from X to Y . Given two arrows $f, g \in \mathbf{C}[X, Y]$, $\alpha : f \Rightarrow g$ is called a *2-cells* with source f and target g .

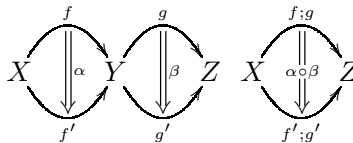
A *2-category* is a category where every homset is the class of objects of some category $\mathbf{C}[X, Y]$ and whose composition functions $\mathbf{C}[X, Y] \times \mathbf{C}[Y, Z] \rightarrow \mathbf{C}[X, Z]$ are functors associative and with identity. Explicitly, a 2-category is a category enriched with 2-cells, such that:

- For every arrow $f \in \mathbf{C}[X, Y]$ there is an identity 2-cell $1_f : f \Rightarrow f$.
- For every pair of 2-cells $\alpha : f \Rightarrow g$ and $\beta : g \Rightarrow h$ is defined $\alpha \bullet \beta : f \Rightarrow h$. Moreover \bullet is associative and with identity.
- For every pair of 2-cells $\alpha : f \Rightarrow f'$ and $\beta : g \Rightarrow g'$ such that $f, f' \in \mathbf{C}[X, Y]$ and $g, g' \in \mathbf{C}[Y, Z]$ is defined $\alpha \circ \beta : f; g \Rightarrow f'; g'$. Moreover \circ is associative and has as identity 1_{id_X} .
- For every $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ we have that $1_f \circ 1_g = 1_{f;g}$.
- For every $f, f', f'' \in \mathbf{C}[X, Y]$, $g, g', g'' \in \mathbf{C}[Y, Z]$, $\alpha : f \Rightarrow f'$, $\alpha' : f' \Rightarrow f''$, $\beta : g \Rightarrow g'$ and $\beta' : g' \Rightarrow g''$, we have that $(\alpha \bullet \alpha') \circ (\beta \bullet \beta') = (\alpha \circ \beta) \bullet (\alpha' \circ \beta')$.

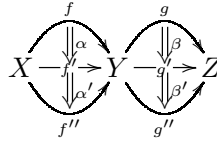
The first two points just say that every homset is a category, i.e., there exists the identity arrows (that, in this case, are 2-cells) and there is a composition operation between arrows that is associative and with identity. This operation is called *vertical composition* and denoted by \bullet .



The last three points just correspond to say that the composition functions are functors. Here we denote by $;$ the composition between arrows (i.e., objects of the homsets) and by \circ the *horizontal composition* of 2-cells.



The last point is called *middle four interchange law*. It states that in 2-categories the order of composition of 2-cells is not important. Indeed it can be shown that in any possible diagram, all the possible different ways to combine together vertical and horizontal composition, yield the same composite 2-cell. This primitive operation is called *pastings*.



The canonical example of 2-category is **Cat** the category of categories, functors and natural transformations [12].

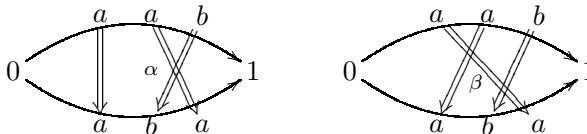
A *G-Category* is a 2-category where all the 2-cells are isomorphisms.

3 The Theory of G-Reactive Systems

Reactive System *à la* Leifer and Milner [14] are defined on a (Lawvere like) category **C** that represents the syntax of some formalism. Arrows of **C** are terms and contexts, while objects are types. More precisely, every arrows represents a class of structurally equivalent terms (or contexts), i.e., all terms are considered up to structural equivalence (\equiv). The theory allows to derive from reduction rules a labeled transition system, and if **C** has a special colimit, called relative pushout (RPO), then bisimilarity is a congruence. Sassone and Sobociński have shown that quotienting the syntax up to structural equivalence usually brings to a category where there are not RPOs (Example 2.2.2 in [19]). Moreover considering terms up to structural equivalence usually makes to derive wrong labeled transition system (Example 2.2.1 in [19]).

For these reasons Sassone and Sobociński introduced G-Reactive Systems [17] that are Reactive Systems *à la* Leifer and Milner but defined on a G-category. Here, every arrows is a single term (not up to \equiv) and between two arrows representing structurally equivalent terms, there is a 2-cells that represents a proof of equivalence of the two terms.

As an example, suppose to have a set of constants N and a binary operator \otimes that is associative and commutative. Let **C** be the Lawvere category where the operator is just associative. Arrows of $\mathbf{C}[0, 1]$ are strings on N . Thus the string $a \otimes a \otimes b \equiv a \otimes b \otimes a$, but they are two different arrows in the category **C**. However we can construct a G-category, where the 2-cells are permutations of constants on N . For example there exist the 2-cells $\alpha, \beta : a \otimes a \otimes b \Rightarrow a \otimes b \otimes a$ as depicted below.



The theory is centered on the concepts of *term*, *context* and *reaction rules*: contexts are arrows of a G-category, terms are arrows having as domain 0 (a special object that denotes no holes), and reaction rules are pairs of terms.

Definition 3.1 (G-Reactive System) A *G-Reactive System* \mathcal{R} consists of:

- (i) a G-category **C**;
- (ii) a distinguished object $0 \in |\mathbf{C}|$;

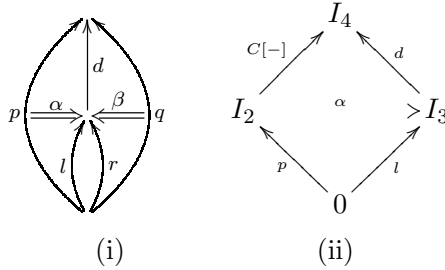


Fig. 1. (i) Reaction; (ii) Redex Square

- (iii) a collection \mathbf{D} of arrows of \mathbf{C} called *reactive contexts*; this is closed under 2-cells and reflect composition;
- (iv) a set of pairs $\mathbb{R} \subseteq \bigcup_{I \in |\mathbf{C}|} \mathbf{C}[0, I] \times \mathbf{C}[0, I]$ of *reaction rules*.

The reactive contexts are those in which a reaction can occur. By composition-reflecting we mean that $d; d' \in \mathbf{D}$ implies $d, d' \in \mathbf{D}$, while the closure property means that given $d \in \mathbf{D}$ and $\alpha : d \Rightarrow d'$ then $d' \in \mathbf{D}$. The latter corresponds to the natural requirement that the class of reactive contexts have to be closed under structural equivalence.

Note that the rules have to be ground, i.e., left-hand and right-hand sides have to be terms without holes and, moreover, with the same codomain.

From reaction rules one generates the reaction relation by closing them under all reactive contexts. Formally the *reaction relation* is defined by taking $p \rightsquigarrow q$ if there is $\langle l, r \rangle \in \mathbb{R}$ and $d \in \mathbf{D}$ such that there exist $\alpha : p \Rightarrow l; d$ and $\beta : q \Rightarrow r; d$. This is depicted in Fig. 1(i).

Thus the behaviour of a Reactive System is expressed as an unlabelled transition system. On the other hand many useful behavioural equivalences are only defined for LTSs. In order to obtain an LTS, we can plug a term p into some context $C[-]$ and observe if a reaction occurs. In this case we have that $p \xrightarrow{C[-]}$. Categorically speaking this means that there exists $\alpha : p; C[-] \Rightarrow l; d$ for some rule $\langle l, r \rangle \in \mathbb{R}$ and some reactive context d . This situation is formally depicted by diagram (ii) in Figure 1: a commuting diagram like this is called a *redex square*.

Definition 3.2 (context transition system) The *context transition system* (*CTS* for short) is defined as follows:

- states: arrows $p : 0 \rightarrow I$ in \mathbf{C} , for arbitrary I ;
- transitions: $p \xrightarrow{C[-]} q$ iff $C[p] \rightsquigarrow q$.

Note that this labelled transition system is often infinite-branching since all contexts that allow reactions may occur as labels. Another problem of *CTS* is that it has redundant transitions. For example, consider the term $a.0$ of CCS. The observer can put this term into the context $\bar{a}.0 \mid -$ and observe a reaction. This corresponds to the transition $a.0 \xrightarrow{\bar{a}.0} 0 \mid 0$. However we also have $a.0 \xrightarrow{p \mid \bar{a}.0} p \mid 0 \mid 0$ as a transition, yet p does not contribute to the reaction. Hence we need a notion of

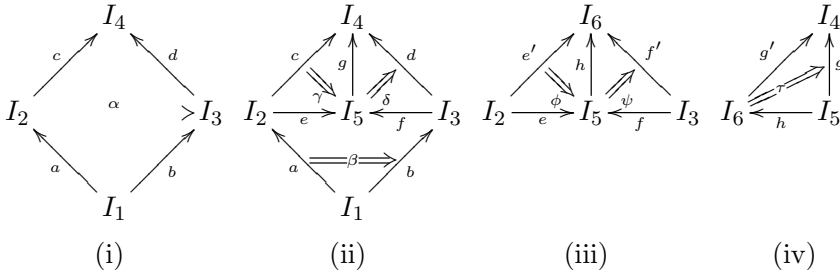


Fig. 2. RPO

“minimal context that allows a reaction”. Leifer and Milner define idem pushouts (IPOs) to capture this notion. Here we introduce the corresponding G-categorical notion as defined by Sassone and Sobociński.

Definition 3.3 (GRPO) Let the diagrams in Figure 2 be in some G-category \mathbf{C} . Any tuple $\langle I_5, e, f, g, \beta, \gamma, \delta \rangle$ as illustrated in diagram (ii) such that $(1_a \circ \gamma) \bullet (\beta \circ 1_g) \bullet (1_b \circ \delta) = \alpha$ is called a *candidate* for (i). In other words, the 2-cells β, γ, δ paste together to give α . A *groupoidal relative pushout (GRPO)* is the smallest such candidate. More formally, it satisfies the universal property that given any other candidate $\langle I_6, e' : I_2 \rightarrow I_6, f' : I_3 \rightarrow I_6, g' : I_6 \rightarrow I_4, \beta' : a; e' \Rightarrow b; f', \gamma' : c \Rightarrow e'; g', \delta' : f'; g \Rightarrow d \rangle$, there exists a *mediating morphism*, i.e., a tuple $\langle h : I_5 \rightarrow I_6, \phi : e' \Rightarrow e; h, \psi : f; h \Rightarrow f', \tau : h'; g' \Rightarrow g \rangle$ illustrated in diagram (iii) and (iv) such that:

- (i) $\gamma = \gamma' \bullet (\phi \circ 1_{g'}) \bullet (1_e \circ \tau)$;
- (ii) $\delta = (id_f \circ \tau^{-1}) \bullet (\psi \circ id_{g'}) \bullet \delta'$;
- (iii) $\beta' = (id_a \circ \phi) \bullet (\beta \circ 1_h) \bullet (id_b \circ \psi)$.

Moreover it must be *essentially unique*, i.e., for any other mediating morphism $\langle h', \phi', \psi', \tau' \rangle$ there must exist a unique 2-cell $\epsilon : h \Rightarrow h'$ which makes the two mediating morphisms compatible, i.e.:

- (i) $\phi' = \phi \bullet (id_e \circ \epsilon)$;
- (ii) $\psi' = (id_f \circ \epsilon^{-1}) \bullet \psi$;
- (iii) $\tau = (\epsilon \circ id_{g'}) \bullet \tau'$.

Definition 3.4 (GIPO) A commuting square such as diagram (i) of Figure 2 is called *idem pushout (GIPO)* if $\langle I_4, c, d, id_{I_4}, \alpha, 1_c, 1_d \rangle$ is its GRPO.

We say that a Reactive System *has GRPOs* if, in the underlying category, for each $\alpha : a; c \Rightarrow b; d$ there exists a GRPO, while we say that it *has redex GRPOs*, if for each redex square there exists a GRPO.

Definition 3.5 (GIPO LTS) The *GIPO LTS (GLTS for short)* is defined as follows:

- states: $p : 0 \rightarrow I$ in \mathbf{C} , for arbitrary I ;

- transitions: $p \xrightarrow{C[-]}_G r; d$ iff there exist $d \in \mathbf{D}$, $\langle l, r \rangle \in \mathbb{R}$ and $\alpha : p; C[-] \Rightarrow l; d$ such that diagram (ii) in Figure 1 is a GIPO.

In other words, if inserting p into the context $C[-]$ matches $l; d$, and $C[-]$ is the “smallest” such context (according to the GIPO condition), then p transforms to $r; d$ with label $C[-]$, where r is the reduct of l . The *GLTS* is called *concrete LTS* in [17], since they introduce also an abstract version, where states and labels are quotiented up to \equiv . In this paper we provide a coalgebraic characterization only of the concrete LTS, since the abstract version is completely analogous.

Bisimilarity on *GLTS* is referred to as *concrete bisimilarity* (denoted by \sim_G). Sassone and Sobociński have shown that if the Reactive System has redex GRPOs, then it is a congruence (i.e., it is preserved under all contexts).

Proposition 3.6 *Let \mathcal{R} be a Reactive System having redex GRPOs, then \sim_G is a congruence.*

4 Coalgebras and Structured Coalgebras

In this section we first introduce the standard way to represent labeled transition systems as coalgebras for a suitable powerset functor [15], and then we discuss how this encoding can be lifted to a more structured framework, where the coalgebraic representation keeps the relevant algebraic structure of the states and transition of the encoded system. Let us start introducing the formal definition of coalgebra for a functor.

Definition 4.1 (coalgebras) Let $\mathbf{B} : \mathbf{C} \rightarrow \mathbf{C}$ be an endofunctor on a category \mathbf{C} . A *coalgebra for \mathbf{B}* or *\mathbf{B} -coalgebra* is a pair $\langle A, a \rangle$ where A is an object of \mathbf{C} and $a : A \rightarrow \mathbf{B}(A)$ is an arrow. A *\mathbf{B} -cohomomorphism* $f : \langle A, a \rangle \rightarrow \langle A', a' \rangle$ is an arrow $f : A \rightarrow A'$ of \mathbf{C} such that

$$f; a' = a; \mathbf{B}(f). \quad (1)$$

The category of \mathbf{B} -coalgebras and \mathbf{B} -cohomomorphisms will be denoted by $\mathbf{Coalg}_{\mathbf{B}}$. The *underlying functor* $\mathbf{U} : \mathbf{Coalg}_{\mathbf{B}} \rightarrow \mathbf{C}$ maps an object $\langle A, a \rangle$ to A and an arrow f to itself.

Let $\mathbf{P}_L : \mathbf{Set} \rightarrow \mathbf{Set}$ be the functor defined as $X \mapsto \mathbf{P}(L \times X)$ where L is a fixed set of labels and \mathbf{P} denotes the powerset functor. Then coalgebras for this functor are one-to-one with labeled transition systems over L [15].

Definition 4.2 (labeled transition systems) Let L be a fixed set of labels. A (*nondeterministic*) *labeled transition system (over L)*, briefly *LTS*, is a structure $TS = \langle S, \longrightarrow_{TS} \rangle$, where S is a set of states, and $\longrightarrow_{TS} \subseteq S \times L \times S$ is a labeled transition relation. As usual, we write $s \xrightarrow{l}_{TS} s'$ for $\langle s, l, s' \rangle \in \longrightarrow_{TS}$.

A *transition system morphism* $f : TS \rightarrow TS'$ is a function $f : S \rightarrow S'$ which “preserves” the transitions, i.e., such that $s \xrightarrow{l}_{TS} t$ implies $f(s) \xrightarrow{l}_{TS'} f(t)$. We will denote by \mathbf{LTS}_L the category of finitely-branching LTS over L and corresponding morphisms.

Proposition 4.3 (labeled transition systems as coalgebras) *Category $\mathbf{Coalg}_{\mathbf{P}_L}$ is isomorphic to the sub-category of \mathbf{LTS}_L containing all its objects, and all the morphisms $f : TS \rightarrow TS'$ which also “reflect” transitions, i.e., such that if $f(s) \xrightarrow{l}_{TS'} t$ then there is a state $s' \in S$ such that $s \xrightarrow{l}_{TS} s'$ and $f(s') = t$.*

It is instructive to spell out the correspondence just stated. For objects, a transition system $\langle S, \longrightarrow \rangle$ is mapped to the coalgebra $\langle S, \sigma \rangle$ where $\sigma(s) = \{\langle l, s' \rangle \mid s \xrightarrow{l} s'\}$, and, vice versa, a coalgebra $\langle S, \sigma : S \rightarrow P_L(S) \rangle$ is mapped to the system $\langle S, \longrightarrow \rangle$, with $s \xrightarrow{l} s'$ if $\langle l, s' \rangle \in \sigma(s)$. For arrows, by spelling out condition (1) for functor \mathbf{P}_L , we get

$$\forall s \in S, \{ \langle l, t \rangle \mid f(s) \xrightarrow{l} t \} = \{ \langle l, f(s') \rangle \mid s \xrightarrow{l} s' \},$$

and by splitting this set equality in the conjunction of the two inclusions, one can easily see that inclusion “ \supseteq ” is equivalent to $s \xrightarrow{l} s' \Rightarrow f(s) \xrightarrow{l} f(s')$, showing that f is a transition system morphism, while the left-to-right inclusion is equivalent to $f(s) \xrightarrow{l} t \Rightarrow \exists s'. s \xrightarrow{l} s' \wedge f(s') = t$, meaning that f is a “zig-zag” morphism, i.e., that it reflects transitions.

The property of “reflecting behaviors” enjoyed by cohomomorphisms is pivotal, for example, in the characterization of bisimulation relations as spans of cohomomorphisms, in the relevance of final coalgebras, and in various other results of the theory of coalgebras [15]. Given two coalgebras $\langle A, a \rangle$ and $\langle A', a' \rangle$, a *coalgebraic bisimulation* on them is a relation $R \subseteq A \times A'$ such that $\langle R, r \rangle$ is a coalgebra and the projections $\pi : R \rightarrow A$ and $\pi' : R \rightarrow A'$ are cohomomorphisms. Interestingly, it is easy to check that two states of a labeled transition system S are bisimilar (in the standard sense) if and only if there is a coalgebraic bisimulation $R \subseteq S \times S$ (regarded as a \mathbf{P}_L -coalgebra) which relates them.

An even easier definition of categorical bisimilarity can be given if there exists a *final coalgebra*. In this case, two elements of the carrier of a coalgebra are bisimilar iff they are mapped to the same element of the final coalgebra by the unique cohomomorphism. Unfortunately, due to cardinality reasons, the functor \mathbf{P}_L used for the coalgebraic representation of transition systems does not admit a final coalgebra [15]. One satisfactory, alternative solution consists of replacing the powerset functor \mathbf{P} on \mathbf{Set} by the *countable* powerset functor \mathbf{P}_c , which maps a set to the family of its countable subsets. Then defining the functor $\mathbf{P}_L^c : \mathbf{Set} \rightarrow \mathbf{Set}$ by $X \mapsto \mathbf{P}_c(L \times X)$ one has that coalgebras for this endofunctor are in one-to-one correspondence with transition systems with *countable degree*, i.e., systems where for each state $s \in S$ the set $\{ \langle s', l \rangle \mid s \xrightarrow{l} s' \}$ is countable. Unlike functor \mathbf{P}_L , the functor \mathbf{P}_L^c admits cofree and final coalgebras (Example 6.8 of [15]).

Proposition 4.4 (final and cofree \mathbf{P}_L^c -coalgebras) *The obvious underlying functor $\mathbf{U} : \mathbf{Coalg}_{\mathbf{P}_L^c} \rightarrow \mathbf{Set}$ has a right adjoint $\mathbf{R} : \mathbf{Set} \rightarrow \mathbf{Coalg}_{\mathbf{P}_L^c}$ associating with each set X a cofree coalgebra over X . As a consequence, the category $\mathbf{Coalg}_{\mathbf{P}_L^c}$ has a final object, which is the cofree coalgebra $\mathbf{R}(1)$ over a final set 1 .*

We shall stick to this functor throughout the rest of the paper, and since there is

no room for confusion the superscript c will be understood.

For Reactive Systems, as well as process algebra and tile rewrite systems, the coalgebraic representation using functor \mathbf{P}_L (for a suitable L) introduced in Proposition 4.3 is not completely satisfactory, because by definition the carrier is just a set and therefore the algebraic structure of states is lost. This calls for the introduction of *structured coalgebras*, i.e., coalgebras for an endofunctor on a category \mathbf{Alg}_Γ of algebras for a signature (or algebraic specification) Γ which is determined by the structure of states. Since it is natural to require that the structured coalgebraic representation of a system is compatible with the unstructured, set-based one, the following notion will be relevant.

Definition 4.5 (lifting) Given endofunctors $\mathbf{B} : \mathbf{C} \rightarrow \mathbf{C}$, $\mathbf{B}' : \mathbf{C}' \rightarrow \mathbf{C}'$ and a functor $\mathbf{V} : \mathbf{C}' \rightarrow \mathbf{C}$, \mathbf{B}' is called a *lifting of \mathbf{B} along \mathbf{V}* , if $\mathbf{B}'\mathbf{V} = \mathbf{V}\mathbf{B}$.

$$\begin{array}{ccc} \mathbf{C}' & \xrightarrow{\mathbf{B}'} & \mathbf{C}' \\ \mathbf{V} \downarrow & & \downarrow \mathbf{V} \\ \mathbf{C} & \xrightarrow{\mathbf{B}} & \mathbf{C} \end{array}$$

In particular, if $\mathbf{V}^\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Set}$ is the underlying set functor, one will consider typically a functor $\mathbf{B}' : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Alg}_\Gamma$ which is a lifting of \mathbf{P}_L along \mathbf{V}^Γ .

The structured coalgebraic representation of transition systems has been studied in [20] for the case of CCS and other process algebra whose operational semantics is given by SOS rules in the DeSimone format, and in [9] for tile systems. In the first case the lifting of \mathbf{P}_L is determined by the SOS rules, while in the second one it is defined by authors'hand. In both cases, as well as for the case of G-Reactive Systems addressed in the next sections, the following interesting fact applies [20,7].

Proposition 4.6 (bisimilarity is a congruence in structured coalgebras) Let Γ be an algebraic specification, and $\mathbf{B}'_\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Alg}_\Gamma$ be a lifting of $\mathbf{P}_L : \mathbf{Set} \rightarrow \mathbf{Set}$. If $\langle S, \sigma \rangle$ is a \mathbf{B}'_Γ -coalgebra and $\langle S, \longrightarrow \rangle$ its corresponding structured LTS, then bisimilarity on $\langle S, \longrightarrow \rangle$ is a congruence with respect to the operators in Γ .

The statement follows by the observation that the right adjoint $R : \mathbf{Set} \rightarrow \mathbf{Coalg}_{\mathbf{P}_L}$ of Proposition 4.4 lifts to a right adjoint $\mathbf{R}^\Gamma : \mathbf{Alg}_\Gamma \rightarrow \mathbf{Coalg}_{\mathbf{B}'_\Gamma}$ for the forgetful functor \mathbf{U}^Γ , with $\mathbf{V}^\Gamma; \mathbf{R} = \mathbf{R}^\Gamma; \mathbf{V}'_\Gamma$ (see [20]), as shown in the following diagram.

$$\begin{array}{ccc} \mathbf{Coalg}_{\mathbf{P}_L} & \xrightarrow{\mathbf{F}'_\Gamma} & \mathbf{Coalg}_{\mathbf{B}'_\Gamma} \\ \mathbf{R} \uparrow \downarrow \mathbf{U} & & \mathbf{R}^\Gamma \uparrow \downarrow \mathbf{U}^\Gamma \\ \mathbf{Set} & \xrightarrow{\mathbf{F}^\Gamma} & \mathbf{Alg}_\Gamma \\ & \xleftarrow{\mathbf{V}^\Gamma} & \end{array}$$

Now, since \mathbf{R}^Γ and \mathbf{V}'_Γ are both right adjoints, $\mathbf{Coalg}_{\mathbf{B}'_\Gamma}$ inherits a final object $\mathbf{R}^\Gamma(\mathbf{1})$ from \mathbf{Alg}_Γ which is then preserved by \mathbf{V}'_Γ . Hence, bisimilarity induced by the final morphism to $\mathbf{R}^\Gamma(\mathbf{1})$ in $\mathbf{Coalg}_{\mathbf{B}'_\Gamma}$ is determined by the underlying sets and

functions, that is, its definition does not use the algebraic structure of states and transitions. Since the final morphisms in $\mathbf{Coalg}_{\mathbf{B}_\Gamma}$ are Γ -homomorphisms, it follows that bisimilarity is a congruence.

In other words, a transition system can be represented as a structured coalgebra only if bisimilarity is a congruence. This property certainly holds, for example, for specifications in GSOS format, which are considered in [20]. Certain structures are used there, called *bialgebras*, which combine aspects of algebras and coalgebras: bialgebras can be regarded as an alternative, equivalent presentation of structured coalgebras [7]. A specification in GSOS format is shown to satisfy a certain diagram called *pentagonal law*, which ensures the existence both of an algebra of transition systems and of an algebraic structure on their states. The pentagonal law also makes sure that bisimilarity is a congruence, showing that GSOS specifications perfectly fit in the structured coalgebraic framework.

5 G-Reactive Systems as Coalgebras

In this section we give a coalgebraic characterization of GIPO Labeled Transition System of G-Reactive Systems through the theory outlined in the previous section.

First of all, we have to define the universe of observations. Since the labels of the *GLTS* are arrows of the base G-category \mathbf{C} (representing the contexts), we define the functor as parametric w.r.t. \mathbf{C} , and $\|\mathbf{C}\|$ (i.e. the class of all arrows of \mathbf{C}) is the universe of labels.

Definition 5.1 Given a G-category \mathbf{C} , the functor $\mathbf{P}_{\mathbf{C}} : \mathbf{Set}^{|\mathbf{C}|} \rightarrow \mathbf{Set}^{|\mathbf{C}|}$ is defined for each $|\mathbf{C}|$ -indexed set S by $\mathbf{P}_{\mathbf{C}}(S_n) = \mathbf{P}_c \left(\bigcup_{m \in |\mathbf{C}|} \mathbf{C}[n, m] \times S_m \right)$.

The functor is defined analogously on arrows of $\mathbf{Set}^{|\mathbf{C}|}$.

Note that $\mathbf{P}_{\mathbf{C}}$ is not an endofunctor on \mathbf{Set} , as it is the case for the standard $\mathbf{P}_{\mathbf{L}}$ discussed in the previous section, but it is defined on $\mathbf{Set}^{|\mathbf{C}|}$, i.e., the G-category of sets indexed by objects of \mathbf{C} . The base G-category \mathbf{C} induces $\overline{\mathbf{C}}$, an object of $\mathbf{Set}^{|\mathbf{C}|}$ where, for any sort n , the corresponding set is $\mathbf{C}[0, n]$. Here we have implicitly assumed that \mathbf{C} is *locally small* (i.e., the hom-class between two objects is always a set and not a proper class), otherwise $\mathbf{C}[0, n]$ could be a proper class. Moreover, in the following definition, we require that $\|\mathbf{C}\|$ is a countable set, otherwise the possible transitions of an element could be uncountable and then not belonging to $\mathbf{P}_{\mathbf{C}}$. Note that this usually holds in those categories where arrows are syntactic contexts of a formalism.

Definition 5.2 Given a G-Reactive System $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$, the $\mathbf{P}_{\mathbf{C}}$ -coalgebra corresponding to its *GLTS* is $\langle \overline{\mathbf{C}}, \alpha_{\mathcal{R}}^{\mathbf{C}} \rangle$ where

$$\alpha_{\mathcal{R}}^{\mathbf{C}}(p) = \{(c, r; d) \text{ s.t. diagram (ii) in Fig. 1 is a GIPO and } d \in \mathbf{D} \text{ and } \langle l, r \rangle \in \mathbb{R}\}.$$

It is immediate to see that the LTS defined above exactly coincides with the *GLTS* (Def. 3.5).

From the above definition immediately follows a characterization of \sim_G as final semantics. Indeed the category $\mathbf{Coalg}_{\mathbf{P}_{\mathbf{C}}}$ has a final object $1_{\mathbf{Coalg}_{\mathbf{P}_{\mathbf{C}}}}$, and thus there is a unique morphism $!_{\mathcal{R}} : \langle \overline{\mathbf{C}}, \alpha_{\mathcal{R}}^{\mathbf{C}} \rangle \rightarrow 1_{\mathbf{Coalg}_{\mathbf{P}_{\mathbf{C}}}}$. The whole theory of coalgebra assures that $!_{\mathcal{R}}(p) = !_{\mathcal{R}}(q)$ if and only if $p \sim_G q$.

6 Lifting the Categorical Structure

The coalgebraic model defined in the previous section does not take into account the algebraic structure of the states, i.e., of the possibility of contextualizing a term. In order to have a richer model we lift that construction to a structured setting where the base category is not anymore $\mathbf{Set}^{|\mathbf{C}|}$, but a category of algebras with contextualization operations. In the following we assume that the category \mathbf{C} has *strict distinguished object*, i.e., that the only arrow with target 0 is id_0 . This is needed to distinguish between elements and operations of algebras.

specification $\Gamma(\mathcal{R}) =$

sorts

$$n \quad \forall n \in |\mathbf{C}| \text{ with } n \neq 0$$

operations

$$d : n \rightarrow m \quad \forall d \in \mathbf{C}[n, m] \text{ with } n \neq 0$$

This specification defines $\mathbf{Alg}_{\Gamma(\mathcal{R})}$ the category of $\Gamma(\mathcal{R})$ -algebras and $\Gamma(\mathcal{R})$ -homomorphisms. The base category \mathbf{C} of a Reactive System induces $\widehat{\mathbf{C}} \in |\mathbf{Alg}_{\Gamma(\mathcal{R})}|$. In $\widehat{\mathbf{C}}$, for every sort m , the elements of this sort are the arrows of $\mathbf{C}[0, m]$. Every operation $c : m \rightarrow n$ is defined for every element p of sort m as the composition of $p; c$ in \mathbf{C} . Note that we removed arrows with source 0 from the operations, otherwise all the elements of $\widehat{\mathbf{C}}$ must be considered as constants and not as elements.

Hereafter we will use $c_{\mathfrak{X}}$ to denote the operation c of the algebra \mathfrak{X} , and c to mean both the operation $c_{\widehat{\mathbf{C}}}$ and the arrow $c \in \|\mathbf{C}\|$. Moreover we will not specify the sort of sets and operations, in order to make the whole presentation more readable.

Definition 6.1 The functor $\mathbf{F} : \mathbf{Alg}_{\Gamma(\mathcal{R})} \rightarrow \mathbf{Alg}_{\Gamma(\mathcal{R})}$ is defined as follows.

For each $\mathfrak{X} = \langle X, a_{\mathfrak{X}}, b_{\mathfrak{X}}, \dots \rangle \in \mathbf{Alg}_{\Gamma(\mathcal{R})}$, $\mathbf{F}(\mathfrak{X}) = \langle \mathbf{P}_{\mathbf{C}}(X), a_{\mathbf{F}(\mathfrak{X})}, b_{\mathbf{F}(\mathfrak{X})}, \dots \rangle$

where $\forall a \in \Gamma(\mathcal{R}), \forall A \in \mathbf{P}_{\mathbf{C}}(X), a_{\mathbf{F}(\mathfrak{X})}(A) =$

$$\{(c, d_{\mathfrak{X}}(x)) \text{ s.t. diagram (i) in Fig. 2 is a GIPO in } \mathbf{C}, d \in \mathbf{D} \text{ and } (b, x) \in A\}.$$

On arrows of $\mathbf{Alg}_{\Gamma(\mathcal{R})}$ is defined as $\mathbf{P}_{\mathbf{C}}$.

Proposition 6.2 Let $\mathbf{V}^{\Gamma(\mathcal{R})} : \mathbf{Alg}_{\Gamma(\mathcal{R})} \rightarrow \mathbf{Set}^{|\mathbf{C}|}$ be the forgetful functor that associates to each $\Gamma(\mathcal{R})$ -algebra its carrier set. Then \mathbf{F} is a lifting of $\mathbf{P}_{\mathbf{C}}$ along $\mathbf{V}^{\Gamma(\mathcal{R})}$.

Proof. For any $\mathfrak{X} \in |\mathbf{Alg}_{\Gamma(\mathcal{R})}|$ with carrier X , $\mathbf{V}^{\Gamma(\mathcal{R})}(\mathbf{F}(\mathfrak{X})) = \mathbf{P}_{\mathbf{C}}(X)$. Moreover $\mathbf{V}^{\Gamma(\mathcal{R})}(\mathfrak{X}) = X$ and then $\mathbf{P}_{\mathbf{C}}(\mathbf{V}^{\Gamma(\mathcal{R})}(\mathfrak{X})) = \mathbf{P}_{\mathbf{C}}(X)$.

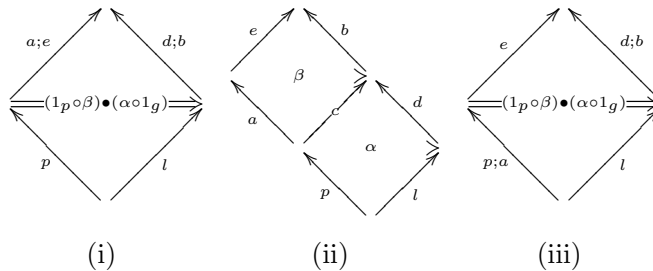
For any $h \in \|\mathbf{Alg}_{\Gamma(\mathcal{R})}\|$, $\mathbf{V}^{\Gamma(\mathcal{R})}(\mathbf{F}(h)) = \mathbf{P}_{\mathbf{C}}(h) = \mathbf{P}_{\mathbf{C}}(\mathbf{V}^{\Gamma(\mathcal{R})}(h))$. \square

Trivially \mathbf{F} is a lifting of $\mathbf{P}_{\mathbf{C}}$. Then, by Prop. 4.6, $\mathbf{Coalg}_{\mathbf{F}}$ is a category of bialgebras, it has final object $1_{\mathbf{Coalg}_{\mathbf{F}}}$ and bisimilarity abstracts away from the algebraic structure.

In [20], Turi and Plotkin show that every process algebra whose operational semantics is given by SOS rules in DeSimone format, defines a bialgebra. In that approach the carrier of the bialgebra is an initial algebra T_{Σ} for a given algebraic signature Σ , and the SOS rules in DeSimone format specify how an endofunctor \mathbf{F}_{Σ} behaves with respect to the operations of the signature. Since there exists only one arrow $?_{\Sigma} : T_{\Sigma} \rightarrow \mathbf{F}_{\Sigma}(T_{\Sigma})$, to give the SOS rules is enough for defining a bialgebra (i.e., $\langle T_{\Sigma}, ?_{\Sigma} \rangle$) and then for assuring compositionality of bisimilarity. Our construction slightly differs from this. Indeed, the carrier of our coalgebra is $\widehat{\mathbf{C}}$, that is not the initial algebra of $\mathbf{Alg}_{\Gamma(\mathcal{R})}$. Then there could exist several or no structured coalgebras with carrier $\widehat{\mathbf{C}}$. In the following we prove that $\alpha_{\mathcal{R}}^{\mathbf{C}} : \widehat{\mathbf{C}} \rightarrow \mathbf{F}(\widehat{\mathbf{C}})$ is a $\Gamma(\mathcal{R})$ -homomorphism. This automatically assures that $\langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}}^{\mathbf{C}} \rangle$ is a structured coalgebra and then bisimilarity is a congruence with respect to the operations of $\Gamma(\mathcal{R})$.

Theorem 6.3 *Let $\mathcal{R} = \langle \mathbf{C}, 0, \mathbf{D}, \mathbb{R} \rangle$ be a Reactive System with redex-RPOs. If $\|\mathbf{C}\|$ is countable and \mathbf{C} has strict distinguished object, then $\langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}}^{\mathbf{C}} \rangle$ is a \mathbf{F} -coalgebra.*

Proof. In order to prove that $\langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}}^{\mathbf{C}} \rangle$ is a \mathbf{F} -coalgebra, we have to prove that $\alpha_{\mathcal{R}}^{\mathbf{C}}$ is a $\Sigma(\mathbf{C})$ -homomorphism, i.e., that for all a and p , $a_{\mathbf{F}(\widehat{\mathbf{C}})}(\alpha_{\mathcal{R}}^{\mathbf{C}}(p)) = \alpha_{\mathcal{R}}^{\mathbf{C}}(a_{\widehat{\mathbf{C}}}(p))$.



Suppose that $(e, p') \in a_{\mathbf{F}(\widehat{\mathbf{C}})}(\alpha_{\mathcal{R}}^{\mathbf{C}}(p))$, then there exists two GIPOs as those of diagram (ii) above, such that $p' = r; d; b$ for $\langle l, r \rangle \in \mathbb{R}$ and $d, b \in \mathbf{D}$. Since the two inner squares are GIPOs, also the outer square, diagram (iii), is a GIPO (by composition property of GIPOs, Lemma 2.2.20 of [19]) and then $(e, p') \in \alpha_{\mathcal{R}}^{\mathbf{C}}(a_{\widehat{\mathbf{C}}}(p))$.

Now suppose that $(e, p') \in \alpha_{\mathcal{R}}^{\mathbf{C}}(a_{\widehat{\mathbf{C}}}(p))$, then the diagram (iii) is a GIPO, $d; b \in \mathbf{D}$, $\langle l, r \rangle \in \mathbb{R}$ and $p' = r; d; b$. Since \mathbf{C} has redex-GRPOs, then there exists the GRPO of diagram (i). Let diagram (ii) be such GRPO, then the lower square is a GIPO. By decomposition property of GIPOs (Lemma 2.2.20 of [19]) also the upper square of diagram (ii) is a GIPO. Thus $(c, r; d) \in \alpha_{\mathcal{R}}^{\mathbf{C}}(p)$ and $(e, p') \in c_{\mathbf{F}(\widehat{\mathbf{C}})}(\alpha_{\mathcal{R}}^{\mathbf{C}}(p))$. \square

Note that this proof is completely analogous to that of Theorem 1 of [6] that is the analogous theorem for Reactive System *à la* Leifer and Milner. Here we use composition and decomposition of GIPOs, while there we used composition and decomposition of IPOs. This makes evidence of the fact that it is not important what kind of construction we use, but it is important that the construction satisfies these two property. This is really evident in tile systems [11] where the theory prove

that *decomposition property* guarantees that bisimilarity is a congruence.

From the fact that $\langle \widehat{\mathbf{C}}, \alpha_{\mathcal{R}}^{\mathbf{C}} \rangle$ is a \mathbf{F} -coalgebra immediately follows that \sim_G is a congruence with respect to the operations of $\Gamma(\mathcal{R})$.

7 Conclusions and Future Work

This paper extends [6] to G-Reactive Systems.

Here we have shown that given a G-Reactive System we can always construct a coalgebra equivalent to the distilled GIPO Labeled Transition Systems. In such a way we define a final semantics for Reactive Systems. Moreover if the Reactive Systems has GRPOs, then we can define a structured coalgebra equivalent to the distilled LTS, and then bisimilarity is compositional. This can be seen as a general, more abstract proof, of compositionality of bisimilarity for Reactive Systems. However this is not the only aim of the paper.

Indeed [6] coalgebraically characterizes also *saturated bisimilarity* [5], a coarser equivalence that take into account not only the minimal contexts but all the possible. Since all the possible contexts are usually infinite in number, we introduced *Normalized Coalgebras* that allow to forget about redundant transitions, i.e., transitions that are not meaningful in the bisimulation game. The relation between minimal contexts and redundant transitions is deeply studied in [6]. The resulting notion of bisimulation is asymmetric and recall us open bisimulation [16], asynchronous bisimulation [1] and large bisimulation [2].

The construction presented in this paper can be easily extended to normalized coalgebras in order to capture saturated bisimilarity of G-Reactive Systems.

This is particularly interesting in order to check saturated bisimilarity for Borrowed Contexts rewriting [10]. In our opinion, the standard bisimilarity on the (G)IPO LTS is usually too strict, while saturated seems more appropriate. This is the case of Logic Programming and Open π -calculus as shown in [5], and of Open Petri nets as shown in [6].

Another interesting point concerning Borrowed Contexts is the derivation of an SOS semantics as pointed out in [3]. Indeed reasoning on the derived LTS is usually hard, and it will be more useful having an SOS specification of it. This paper can be seen also as a first step in this direction. Indeed, as shown in [20], GSOS specifications can be seen as the lifting of the endofunctor to the algebraic setting. The lifting that we have shown in Section 6 describes the following rule

$$\frac{p \xrightarrow{b}_G q \quad \text{diagram (i) in Fig. 2 is a GIPO}}{p; a \xrightarrow{c}_G q; d}.$$

References

- [1] Amadio, R. M., I. Castellani and D. Sangiorgi, *On bisimulations for the asynchronous pi-calculus.*, in: *Proc. of CONCUR '96*, LNCS **1119** (1996), pp. 147–162.
- [2] Baldan, P., A. Bracciali and R. Bruni, *Bisimulation by unification.*, in: *Proc. of AMAST '02*, LNCS **2422** (2002), pp. 254–270.

- [3] Baldan, P., H. Ehrig and B. König, *Composition and decomposition of dpo transformations with borrowed context.*, in: *ICGT*, LNCS **4178**, 2006, pp. 153–167.
- [4] Berry, G. and G. Boudol, *The chemical abstract machine*, *Theor. Comp. Sci.* **96** (1992), pp. 217–248.
- [5] Bonchi, F., B. König and U. Montanari, *Saturated semantics for reactive systems.*, in: *Proc. of LICS* (2006), pp. 69–80.
- [6] Bonchi, F. and U. Montanari, *Coalgebraic models for reactive systems*, in: *Proc. of CONCUR '07*, To appear in LNCS.
- [7] Corradini, A., M. Große-Rhode and R. Heckel, *Structured transition systems as lax coalgebras.*, *Electr. Notes in Theor. Comp. Sci.* **11** (1998).
- [8] Corradini, A., R. Heckel and U. Montanari, *From sos specifications to structured coalgebras: How to make bisimulation a congruence.*, *Electr. Notes in Theor. Comp. Sci.* **19** (1999).
- [9] Corradini, A., R. Heckel and U. Montanari, *Tile transition systems as structured coalgebras.*, in: *Proc. of FCT*, 1999, pp. 13–38.
- [10] Ehrig, H. and B. König, *Deriving bisimulation congruences in the DPO approach to graph rewriting.*, in: *Proc. of FoSSaCS '05*, LNCS **2987** (2004), pp. 151–166.
- [11] Gadduci, F. and U. Montanari, “The tile model,” MIT Press, 1999 .
- [12] Godement, R., “Topologie Algébrique et Théorie des Faisceaux,” Hermann, 1958.
- [13] Lane, S. M., “Categories for the Working Mathematician,” Springer, 1972.
- [14] Leifer, J. J. and R. Milner, *Deriving bisimulation congruences for reactive systems.*, in: *Proc. of CONCUR '00*, LNCS **1877** (2000), pp. 243–258.
- [15] Rutten, J. J. M. M., *Universal coalgebra: a theory of systems.*, *Theor. Comp. Sci.* **249** (2000), pp. 3–80.
- [16] Sangiorgi, D., *A theory of bisimulation for the pi-calculus.*, *Acta Inf.* **33** (1996), pp. 69–97.
- [17] Sassone, V. and P. Sobocinski, *Locating reaction with 2-categories.*, *Theor. Comp. Sci.* **333** (2005), pp. 297–327.
- [18] Sassone, V. and P. Sobociński, *Reactive systems over cospans*, in: *Proc. of LICS* (2005), pp. 311–320.
- [19] Sobociński, P., “Deriving process congruences from reaction rules.” Ph.D. thesis (2004).
- [20] Turi, D. and G. D. Plotkin, *Towards a mathematical operational semantics.*, in: *Proc. of LICS* (1997), pp. 280–291.