Saturated LTSs for Adhesive Rewriting Systems*

Filippo Bonchi[1], Fabio Gadducci[2],
Giacoma Valentina Monreale[2], and Ugo Montanari[2]

[1] INRIA Saclay and LIX, École Polytechnique
[2] Dipartimento di Informatica, Università di Pisa

Abstract. G-Reactive Systems (GRSs) are a framework for the derivation of labelled transition systems (LTSs) from a set of unlabelled rules. A label for a transition from A to B is a context $C[-]$ such that $C[A]$ may perform a reaction and reach B. If either all contexts, or just the "minimal" ones, are considered, the resulting LTS is called saturated (GIPO, respectively). The borrowed contexts (BCs) technique addresses the issue in the setting of the DPO approach. Indeed, from an adhesive rewriting system (ARS) a GRS can be defined such that DPO derivations correspond to reactions, and BC derivations to transitions of the GIPO LTS. This paper extends the BCs technique in order to derive saturated LTSs for ARSs, applying it to capture bisimilarity for asynchronous calculi.

1 Introduction

The complexity of the formalisms adopted for the specification of open-ended systems (let them be either Web Services description languages [26], or process calculi for biological systems [11], or...) is putting novel emphasis on the use of reduction semantics for modelling their dynamics. Roughly, the modelling technique is based on a reduction system: a set of states of the device equipped with a binary relation, representing the possible evolutions of the device. The system itself is presented inductively, by instantiating a few reduction rules.

The ease of use of the approach based on reductions led to its increasing adoption, most often in presence of specification languages offering a complex interaction between their operators (i.e., between the components of a compound system). However, a main drawback of reduction-based solutions is poor compositionality, since the dynamic behaviour of arbitrary stand-alone terms can be interpreted only by inserting them in appropriate contexts, where a reduction may take place. It would then be preferable to equip such formalisms with a labelled semantics, providing them with a specification of their dynamics which is based on a set of labelled rules, each one compactly describing some behavioural information/features of the component it is applied to. This is the first step leading to the definition of suitable observational equivalences, abstractly characterising when two systems have the same behaviour, thus allowing the possibility of verifying properties of system compositions.

* Supported by the MIUR Project `SisteR`, and carried out during the first author's tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

H. Ehrig et al. (Eds.): ICGT 2010, LNCS 6372, pp. 123–138, 2010.

The identification of the “right” labels is a difficult task: the assessment of their validity may vary for each formalism, and it is usually left to the ingenuity of the researcher. A case at hand is the calculus of mobile ambients [10]: despite its rapid acceptance in the process calculi literature, and the intense scrutiny it was subject to, the development of a suitable labelled semantics defied the researchers until quite recently [23,30], and it might not be fully settled.

G-Reactive Systems (GRSs) [22,31] are a successful meta-framework addressing the need of observational equivalences for systems specified by unlabelled rules. The key idea for getting an LTS is simple: a label for a transition from state A to state B is a context $C[-]$ such that the composed state $C[A]$, obtained by inserting A into $C[-]$, may perform a reaction and reach B. If either all possible contexts are considered, or just the “minimal” ones, the resulting LTS is called saturated or GIPO, respectively.

The framework proved quite general, covering a wide range of applications [25,16,9], as well as theoretically rich enough to be largely extended over the years. Various notions of equivalence were defined for the resulting LTS, such as saturated [8] and barbed [18,7]. Moreover, the possibility of obtaining *sorted* contexts was investigated [3], adding flexibility to the minimality requirement, for a favourite instance of GRSs, bigraphical reactive systems [25,17].

The borrowed contexts (BCs) technique allows the derivation of LTSs in the setting of the DPO approach. What is noteworthy is that it represents a constructive presentation of GRSs. Indeed, consider a widely adopted generalisation of graph transformation, *adhesive rewriting systems* (ARSs) [21]: for every ARS a GRS can be defined such that DPO derivations correspond to reactions, and BC derivations correspond to transitions of the GIPO LTS.

Despite validating case studies [28], so far the boundaries of the BC framework were less tested and extended. The resulting lack of flexibility is clearly a drawback in terms of BCs usability, yet it casts a shadow also on the novel notions introduced for GRSs: indeed, these are of an eminently prescriptive nature, and they would then benefit (as a sanity check, or for clarifying their meaning) from the constructive recasting offered by the BC setting. Our paper aims at addressing this sort of “technology transfer” among the frameworks: we introduce a technique based on BCs which allows to derive saturated LTSs for ARSs, exploiting it to propose an equivalence based on a notion of barb, for those (saturated) LTSs derived by the BC mechanism. Our proposal is then tested on a case study, drawn from the mold of the visual specification of process calculi.

We believe that interesting challenges, as well as one of the most successful application of the BC synthesis mechanism, arise in the derivation of LTSs for process calculi. The modelling approach is straightforward [6,5]: given a calculus, a graphical encoding is found such that structurally congruent processes are mapped into isomorphic graphs, and the reduction semantics is captured by a set of DPO rules, along the lines of [15,13]. These graphs are amenable to the BC mechanism, and a GRS on processes can be obtained such that the resulting GIPO semantics (hopefully) captures the standard bisimilarity for the calculus at hand, as it happens for strong bisimilarity in synchronous CCS [5].

Unfortunately, for most calculi strong bisimilarity is not the preferred equivalence, since it is too discriminating: this fact holds true especially for the asynchronous ones, such as *asynchronous* CCS. Similarly, the GIPO semantics is often unsatisfactory: as proved e.g. for mobile ambients in [7], a suitable notion of saturated (and barbed) GRS has to be taken into account. The reasoning in the above papers was modelled along the following pattern: after using the BC mechanism to derive the labels, a GRS is reverse-engineered, and the larger array of tools of the GRS framework exploited in order to obtain an LTS and a suitable equivalence for the calculus at hand. The results of our paper allows to skip the derivation of the GRS, and to constructively reason directly on ARSs.

Synopsis. Section 2 surveys with some care the main results on ARSs and GRSs, showing that the former are an instance of the latter: DPO derivations in ARSs correspond to reductions in GRSs, and BC derivations to GIPO transitions. The paper extends this correspondence: Section 4 introduces saturated transitions for ARSs, allowing for an ARS modelling asynchronous CCS (presented in Section 3). The equivalence thus derived with the BC mechanism is strictly contained into asynchronous bisimilarity, hence Section 5 proposes a notion of barb for ARSs: barbs and saturated transitions are able to capture asynchronous bisimilarity. The concluding Section 6 discusses future works.

2 On G-Reactive Systems and Borrowed Contexts

The purpose of this section is to recall, as briefly as possible, the main notions concerning G-reactive systems and borrowed contexts, including a sketch of their relationship, in order to fully understand the developments proposed later on.

2.1 G-Reactive Systems

This section introduces *G-reactive systems* (GRSs) [33,32], an extension of *reactive systems* [22] to *groupoidal enriched categories* (*G-categories*). For our purposes, it suffices to know that (a) *2-categories* are categories equipped with *2-cells* (intuitively, "arrows between arrows") and (b) G-categories are 2-categories where all 2-cells are isomorphisms: we refer the reader to the references above.

A *G*-category $\mathbf{B}$ models the syntax of a formalism. Objects represent system *interfaces* (with 0 the empty one). A *system* with interface I_1 is an arrow $A : 0 \to I_1$: it can be plugged into the *context* $B : I_1 \to I_2$ via arrow composition $A; B$. Given arrows $A, B : I_1 \to I_2$, a 2-cell $\alpha : A \Rightarrow B$ represents an isomorphism (intuitively, a proof of equivalence) between contexts A and B. The semantics is instead given via *reduction rules*: pairs of systems $\langle L, R\rangle$ with the same interface.

Definition 1 (G-Reactive System). *A* G-reactive system $\mathbb{C}$ *consists of*

1. *a G-category* $\mathbf{B}$*;*
2. *a distinguished object* 0*;*
3. *a set* $\mathbf{D} \subseteq \mathbf{B}$ *of 2-cells closed, composition-reflecting* reactive contexts*;*
4. *a set* $\mathfrak{R} \subseteq \bigcup_{I \in |\mathbf{C}|} \mathbf{B}(0, I) \times \mathbf{B}(0, I)$ *of* reduction rules*.*

Intuitively, reactive contexts are those arrows inside which a reduction can occur. By composition-reflecting we mean that $D'; D \in \mathbf{D}$ implies $D, D' \in \mathbf{D}$.

The reduction relation is generated by closing the reduction rules under all reactive contexts and 2-cells. Formally, the *reduction relation* is defined by taking $A \leadsto B$ if there exist $\langle L, R \rangle \in \mathfrak{R}$, $D \in \mathbf{D}$, $\alpha : A \Rightarrow L; D$ and $\alpha' : B \Rightarrow R; D$.

The behaviour of a GRS is given by an unlabelled transition system. In order to obtain a labelled one, we plug a system A into a context C and observe if a reduction occurs. Categorically, this means that $A; C$ is isomorphic to $L; D$ (there is a 2-cell $\alpha : A; C \Rightarrow L; D$) for rule $\langle L, R \rangle$ and reactive context D. This situation is depicted in the diagram in Fig. 1, a *redex square*.

Definition 2 (Saturated Transition System). *Let* $\mathbb{C}$ *be a GRS and* $\mathbf{B}$ *its underlying G-category. The* saturated transition system SLTS($\mathbb{C}$) *is defined as*

- *states: arrows* $A : 0 \to I$ *in* $\mathbf{B}$*, for arbitrary* I*;*
- *transitions:* $A \xrightarrow{C}_{SAT} B$ *if* $A; C \leadsto B$*.*

Bisimilarity on SLTS($\mathbb{C}$) *is referred to as* saturated bisimilarity *(*$\sim^S$*).*

Clearly $\sim^S$ is a congruence, and it is the coarsest bisimulation relation that is so. Unfortunately, SLTS($\mathbb{C}$) is often infinite-branching, since all contexts allowing reductions may occur as labels. Moreover, it has redundant transitions: the CCS process like $a.0$ would have both transitions $a.0 \xrightarrow{\overline{a}.0|-}_{SAT} 0$ and $a.0 \xrightarrow{P|\overline{a}.0|-}_{SAT} P$, yet P does not "concur" to the reduction.

The notion of "minimal context allowing a reduction" is modelled by groupoidal-idem pushouts (GIPOs) in G-categories. We refer the reader to [31]: for our purposes, it suffices to know that such "minimality" can be derived: it will become clearer when introducing its constructive version, *borrowed contexts*.

Definition 3 (GIPO Transition System). *Let* $\mathbb{C}$ *be a GRS and* $\mathbf{B}$ *its underlying G-category. The* GIPO transition system GLTS($\mathbb{C}$) *is defined as*

- *states:* $A : 0 \to I$ *in* $\mathbf{B}$*, for arbitrary* I*;*
- *transitions:* $A \xrightarrow{C}_{GIPO} A'$ *if there is* $D \in \mathbf{D}$*, rule* $\langle L, R \rangle \in \mathfrak{R}$*, and 2-cell* $\alpha : A; C \Rightarrow L; D$ *making the diagram in Fig. 1 a GIPO with* A' *iso to* $R; D$*.*

Bisimilarity on GLTS($\mathbb{C}$) *is referred to GIPO bisimilarity (*$\sim^G$*).*

Under certain conditions (see [33,32] for details), $\sim^G$ is a congruence. Unfortunately, it turns out that in many interesting cases $\sim^G$ is too strict [4]. The first and the last authors together with König introduced *semi-saturated bisimilarity*: an alternative, (in some cases) finitary characterisation of saturated bisimilarity. The theory was developed [8] for standard RSs [22], but it is easily lifted to GRSs.

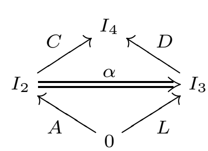


Fig. 1. A GIPO

Definition 4 (Semi-Saturated Bisimulation). *A symmetric relation* $\mathcal{R}$ *is a* semi-saturated bisimulation *if whenever* $A \,\mathcal{R}\, B$ *then*

- *if* $A \xrightarrow{C}_{GIPO} A'$*, then* $B \xrightarrow{C}_{SAT} B'$ *and* $A' \,\mathcal{R}\, B'$*.*

Semi-saturated bisimilarity is shown [8] to coincide with $\sim^S$.

2.2 DPO Rewriting with Borrowed Contexts

This section introduces *double-pushout* (DPO) rewriting and its interactive extension with *borrowed contexts* (BCs) [12]. We present them by relying on adhesive categories [21] as in [33]. In order to uniformly introduce DPO and BCs, we consider DPO derivations for *systems with interface*: morphisms $J \to G$ where G represents a system and J its interface. A *production* or *rewrite rule* is a span $L \leftarrowtail I \to R$ in an adhesive category $\mathbf{A}$ where the left-hand side $I \rightarrowtail L$ is monic. A *DPO adhesive rewriting system* (ARS) is a pair $\langle \mathbf{A}, P\rangle$ where P is a set of productions. In the definitions below, we refer to a chosen ARS $S = \langle \mathbf{A}, P\rangle$.

Definition 5 (DPO Derivation for Systems with Interfaces). *Let $J \to G$ and $J \to H$ be two systems with interface and $p : (L \leftarrowtail I \to R)$ a production. A* match *of p in G is a morphism $m : L \to G$. A* direct derivation *from $J \to G$ to $J \to H$ via p and m is a commuting diagram as in Fig. 2, such that two squares are pushouts (PO), denoted $J \to G \Longrightarrow J \to H$.*

The morphism $k : J \to C$ (making the left triangle commute) is unique, whenever it exists. If such a morphism does not exist, the rewriting step is not feasible.

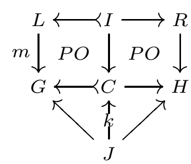


Fig. 2. A direct derivation

In these derivations, the left-hand side L of a production must then occur completely in G. In a BC derivation L might occur partially in G, since the latter may interact with the environment through the interface J in order to exactly match L. Those BCs are the "smallest" contexts needed to obtain the image of L in G, and they may be used as suitable labels. Given an ARS S, $\mathrm{BC}(S)$ denotes the LTS derived via the BC mechanism defined below.

Definition 6 (Rewriting with Borrowed Contexts). *Given a production $p : L \leftarrowtail I \to R$, a system with interface $J \to G$ and a span of monos $d : G \leftarrowtail D \rightarrowtail L$, we say that $J \to G$* reduces to *$K \to H$ with label $J \rightarrowtail F \leftarrow K$* via *$p$ and d if there are objects G^+, C and additional morphisms such that the diagram in Fig. 3 commutes and the squares are either pushouts (PO) or pullbacks (PB). We write $J \to G \xrightarrow{J \rightarrowtail F \leftarrow K} K \to H$, called* rewriting step with borrowed context*.*

The upper left-hand square of the diagram in Fig. 3 merges the left-hand side L and the object G to be rewritten according to a partial match $G \leftarrowtail D \rightarrowtail L$. The resulting G^+ contains a total match of L and is rewritten as in the DPO approach, producing the two other squares in the upper row. The pushout in the lower row gives the BC F which is missing for obtaining a total match of L, along with a morphism $J \rightarrowtail F$ indicating how F should be pasted to G. Finally, the interface for H is obtained by "intersecting" F and C via a pullback.

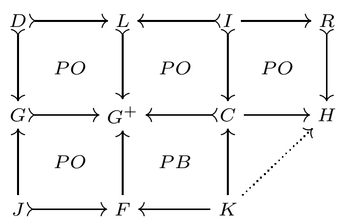


Fig. 3. A BC derivation

Note that two pushout complements that are needed in Definition 6, namely C and F, may not exist. In this case, the rewriting step is not feasible.

2.3 Relating Borrowed Contexts and G-Reactive Systems

We are now ready for showing that ARSs are instances of GRSs [32]. We consider cospans as contexts, and for this reason we need to work in bicategories [2] (with iso 2-cells) instead of G-categories. For the aim of this paper is indeed enough to know that a bicategory can be described, roughly, as a 2-category where associative and identity laws of composition hold up to isomorphism. In order to transfer the notions of GIPOs and GRPOs (in Section 2.1) to bicategories, it suffices to introduce the coherent associativity isomorphisms where necessary.

Bicategories of Cospans. Let $\mathbf{A}$ be an adhesive category with chosen pushouts. The bicategory of cospans of $\mathbf{A}$ has the same objects as $\mathbf{A}$ and morphism pairs $I_1 \xrightarrow{i_C} C \xleftarrow{o_C} I_2$ as arrows from I_1 to I_2, denoted $C_{i_C}^{o_C} : I_1 \to I_2$. Objects I_1 and I_2 are thought of as the input and the output interface of $C_{o_C}^{i_C}$.

Given the cospans $C_{i_C}^{o_C} : I_1 \to I_2$ and $D_{i_D}^{o_D} : I_2 \to I_3$, their composition $C_{i_C}^{o_C}; D_{i_D}^{o_D} : I_1 \to I_3$ is the cospan obtained by taking the chosen pushout of o_C and i_D, as depicted in Fig. 4. Note that, since arrows composition is a chosen pushout, it is associative only up to isomorphism.

A 2-cell $h : C_{i_C}^{o_C} \Rightarrow D_{i_D}^{o_D} : I_1 \to I_2$ is an arrow $h : C \to D$ in $\mathbf{A}$ satisfying $i_C; h = i_D$ and $o_C; h = o_D$, and it is *isomorphic* if h is an isomorphism in $\mathbf{A}$.

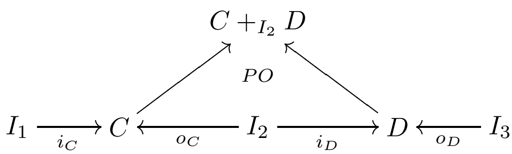


Fig. 4. Cospan composition

A cospan $C_{i_C}^{o_C}$ is *input linear* when i_C is mono in $\mathbf{A}$, and the composition of two input linear cospans yields another input linear cospan. For this reason, we can define the *input linear cospans bicategories* over $\mathbf{A}$, denoted by $ILC(\mathbf{A})$, as the bicategory consisting of input linear cospans and isomorphic 2-cells.

From ARSs to GRSs. Consider an ARS $S = \langle \mathbf{A}, P \rangle$, where the adhesive category $\mathbf{A}$ has initial object 0. This can be seen as a GRS where

- the base category is $ILC(\mathbf{A})$,
- the distinguished object is 0 (the initial object),
- all arrows in $ILC(\mathbf{A})$ are reactive,
- rules are pairs $\langle 0 \to L \leftarrowtail I, 0 \to R \leftarrow I \rangle$ for any $L \leftarrowtail I \to R$ rule in P .

For an ARS S, $\mathbb{C}_S$ denotes its associated GRS. A system with interface $J \to G$ in S can be thought as the arrow $0 \to G \leftarrow J$ of $ILC(\mathbf{A})$.

Proposition 1 ([14]). *$J \to G \Longrightarrow J \to H$ in S iff $J \to G \rightsquigarrow J \to H$ in $\mathbb{C}_S$.*

The above result allows for stating the correspondence between ARSs and BCs: GIPOs for GRSs over input linear cospans are equivalent to BCs for ARSs.

Proposition 2 ([32]). $\textsc{bc}(S) = \textsc{glts}(\mathbb{C}_S)$.

Sections 4 and 5 present the main results of the paper, strengthening the correspondence between the theories. We introduce (barbed) saturated semantics for BCs and we show how it relates to (barbed) saturated semantics for GRSs.

$$p ::= \overline{a},\ p_1 \mid p_2,\ (\nu a)p,\ m \qquad m ::= \mathbf{0},\ \alpha.p,\ m_1 + m_2 \qquad \alpha ::= a,\ \tau$$

$$\begin{array}{lll} p \mid q \equiv q \mid p & (p \mid q) \mid r \equiv p \mid (q \mid r) & p \mid \mathbf{0} \equiv p \\ m + n \equiv n + m & (m+n)+o \equiv m + (n+o) & m+\mathbf{0} \equiv m \\ (\nu a)(\nu b)p \equiv (\nu b)(\nu a)p & (\nu a)(p \mid q) \equiv p \mid (\nu a)q \quad \text{if } a \notin fn(p) & (\nu a)\mathbf{0} \equiv \mathbf{0} \\ & (\nu a)p \equiv (\nu b)(p\{^b/_a\}) \ \text{ if } b \notin fn(p) & \end{array}$$

$$\overline{a} \mid (a.p + m) \to p \qquad \tau.p + m \to p \qquad \frac{p \to q}{(\nu a)p \to (\nu a)q} \qquad \frac{p \to q}{p \mid r \to q \mid r}$$

Fig. 5. Syntax, structural congruence and reduction relation of ACCS

$$a.p + m \xrightarrow{a} p \qquad \tau.p + m \xrightarrow{\tau} p \qquad \bar{a} \xrightarrow{\bar{a}} \mathbf{0}$$

$$\frac{p \xrightarrow{\mu} q \quad a \notin n(\mu)}{(\nu a)p \xrightarrow{\mu} (\nu a)q} \qquad \frac{p \xrightarrow{\mu} q}{p \mid r \xrightarrow{\mu} q \mid r} \qquad \frac{p \xrightarrow{a} p_1 \quad q \xrightarrow{\bar{a}} q_1}{p \mid q \xrightarrow{\tau} p_1 \mid q_1}$$

Fig. 6. Labelled semantics of ACCS

3 Process Semantics via a Graphical Encoding

This section considers asynchronous CCS (ACCS). We present an encoding for the finite fragment of the calculus (adapting the one for the synchronous version in [5]), and we model the reduction semantics of the calculus via a set of DPO rules. Finally, we show that the resulting GIPO semantics is too strict.

Asynchronous CCS. The syntax of ACCS is shown in Fig. 5: $\mathcal{N}$ is a set of *names*, ranged over by $a, b, \ldots$, with $\tau \notin \mathcal{N}$. We let $p, q, \ldots$ range over the set $\mathcal{P}$ of processes and $m, n, \ldots$ over the set $\mathcal{S}$ of summations. With respect to synchronous CCS, the calculus lacks output prefixes: process $\overline{a}$ is thought of as a message, available on a communication media named a, that disappears after its reception. The *free names* $fn(p)$ of a process p are defined as usual.

Processes are taken up to a *structural congruence* (Fig. 5), denoted by $\equiv$. The *reduction relation*, denoted by $\to$, describes process evolution: the least relation $\to \subseteq \mathcal{P} \times \mathcal{P}$, closed under $\equiv$, inductively generated by the rules in Fig.5. The interactive semantics for ACCS is given by the relation over processes up to $\equiv$, obtained by the rules in Fig. 6. We let μ range over the set of labels $\{\tau, a, \bar{a} \mid a \in \mathcal{N}\}$: the names of μ, denoted by $n(\mu)$, are defined as usual. Differently from synchronous calculi, sending messages is non-blocking. Hence, an observer might send messages without knowing about their reception, and inputs are thus deemed as unobservable. This is mirrored in the chosen bisimilarity [1].

Definition 7 (Asynchronous Bisimulation). *A symmetric relation $\mathcal{R}$ is an* asynchronous bisimulation *if whenever $p \,\mathcal{R}\, q$ then*

- *if $p \xrightarrow{\tau} p'$ then $q \xrightarrow{\tau} q'$ and $p' \,\mathcal{R}\, q'$,*
- *if $p \xrightarrow{\bar{a}} p'$ then $q \xrightarrow{\bar{a}} q'$ and $p' \,\mathcal{R}\, q'$,*
- *if $p \xrightarrow{a} p'$ then either $q \xrightarrow{a} q'$ and $p' \,\mathcal{R}\, q'$ or $q \xrightarrow{\tau} q'$ and $p' \,\mathcal{R}\, q' \mid \bar{a}$.*

Asynchronous bisimilarity $\sim^A$ *is the largest asynchronous bisimulation.*

For example, the processes $a.\bar{a} + \tau.\mathbf{0}$ and $\tau.\mathbf{0}$ are asynchronous bisimilar. If $a.\bar{a} + \tau.\mathbf{0} \xrightarrow{a} \bar{a}$, then $\tau.\mathbf{0} \xrightarrow{\tau} \mathbf{0}$ and clearly $\bar{a} \sim^A \mathbf{0} \mid \bar{a}$.

Graphical Encoding for ACCS. For the sake of space, we do not present the formal definition of the graphical encoding of ACCS processes. Indeed, it is analogous to the encoding for processes of the synchronous calculus into typed graphs with interfaces, presented in [5, Definition 9]: it differs only for the choice of the typed graph T_A, depicted in Fig. 7. We remark that choosing a graph typed over T_A means to consider graphs where each node (edge) is labelled by a node (edge) of T_A, and the incoming and outcoming tentacles are preserved.

Intuitively, a graph having as root a node of type $\bullet$ ($\diamond$) corresponds to a process (respectively a summation), while each node of type $\circ$ basically represents a name. Indeed, even if the encoding could be defined by means of operators on typed graphs with interfaces, for ACCS the situation is summed up by saying that a typed graph with interfaces is the encoding of a process P if its underlying graph is almost the syntactic tree of P: each internal node of type $\bullet$ has exactly one incoming edge, except for root, to which an edge labelled *go* is attached.

Going back to the type graph, the edge *rcv* (*snd*) simulates the input prefix (output operator, respectively), while there is no edge for the parallel composition, non-deterministic choice and restriction operators. Edge c is a syntactical device for "coercing" the occurrence of a summation inside a process context, while similarly edge *go* detects the "entry" point of the computation, thus avoiding to perform any reaction below the outermost prefix operators: the latter is needed to properly simulate the reduction semantics of the calculus.

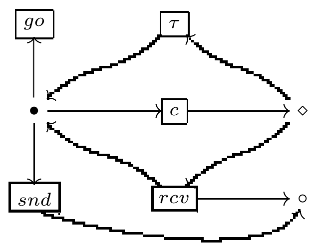


Fig. 7. Type graph T_A

The encoding of a process P, with respect to a set of names Γ including the free names of P, is a graph with interfaces $(\{p\} \cup \Gamma, \emptyset)$. It is sound and complete with respect to structural congruence of the calculus, that is, two processes are equivalent if and only if they are mapped into isomorphic graphs.

Fig. 8 depicts the graph encoding for process $P = (\nu b)(\overline{b} \mid b.\overline{a} + a)$. The two leftmost edges labelled c and *snd* have the same root, into which the node p of the interface is mapped. They are the top edges of the two subgraphs representing the parallel components of the process. In particular, the edge labelled *snd* represents the output over the restricted channel b, namely $\overline{b}$, while the c edge is the syntactical operator denoting that its subgraph represents a summation, that is, $b.\overline{a} + a$. The two leftmost edges of this last subgraph, both labelled *rcv*, model the two input prefixes b and a, while the rightmost *snd* edge represent the operator $\overline{a}$. Note that channel name a is in the interface since it is free in P, while instead the bound name b does not belong to the interface.

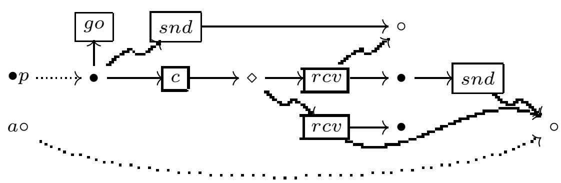


Fig. 8. Encoding for process $(\nu b)(\overline{b} \mid b.\overline{a} + a)$

An ARS for ACCS. Fig. 9 presents the DPO rules which simulate the reduction semantics for ACCS. The two rules p_{com} and p_τ mirror the two axioms of the reductions relation in Fig. 5, and a soundness and completeness result of our encoding with respect to reductions is easily obtained (see [5, Proposition 2]).[1]

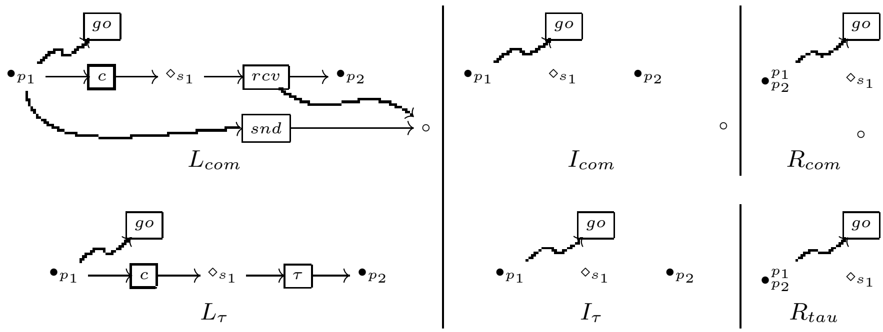


Fig. 9. The productions $p_{com} : L_{com} \leftarrowtail I_{com} \rightarrow R_{com}$ and $p_\tau : L_\tau \leftarrowtail I_\tau \rightarrow R_\tau$

Relevant here is that this graphical encoding is amenable to BC mechanism. By exploiting the pruning techniques of [5], we can show that the derived GIPO transition system is substantially equivalent to the one in Fig. 6: $\xrightarrow{a}$ corresponds to $\rightarrow^{-|\overline{a}}_{GIPO}$, $\xrightarrow{\overline{a}}$ to $\rightarrow^{-|a}_{GIPO}$ and $\xrightarrow{\tau}$ to $\rightarrow^{-}_{GIPO}$, for $-$ the identity context.

However, GIPO bisimilarity is too fine grained for ACCS. Consider the asynchronously bisimilar processes $a.\overline{a} + \tau.\mathbf{0}$ and $\tau.\mathbf{0}$. It is easy to verify that they are not GIPO bisimilar, since the former can perform the GIPO transition $a.\overline{a} + \tau.\mathbf{0} \rightarrow^{-|\overline{a}}_{GIPO} \overline{a} \mid \mathbf{0}$ in Fig. 16 (corresponding to the transition $a.\overline{a} + \tau.\mathbf{0} \xrightarrow{a} \overline{a}$ in the ordinary semantics of Fig. 6), while the latter has no such transition.

4 Saturated Rewriting with Borrowed Contexts

This section introduces a technique based on BCs which allows us to derive saturated LTSs for ARSs. As the BC technique offers a constructive solution for calculating the minimal contexts enabling a rule, the approach we present represents a constructive solution for calculating all contexts enabling a rule.

Definition 8 (Saturated Rewriting). *Given a production $p : L \leftarrowtail I \rightarrow R$ and a system with interface $J \rightarrow G$, we say that $J \rightarrow G$* reduces to *$K \rightarrow H$ with label $J \rightarrowtail F \leftarrow K$* via *$p$ if there are objects G^+, C and additional morphisms such that the diagram in Fig. 10 commutes and three squares are pushouts (PO). We write $J \rightarrow G \xrightarrow{J \rightarrowtail F \leftarrow K}_{BCSAT} K \rightarrow H$, called* saturated rewriting step.

[1] The correspondence accounts for the discarding of sub-processes, due to the solving of non-deterministic choices: after a DPO derivation there can be parts of the graph (representing the discarded components) that are not reachable from the root.

In the diagram on Fig. 10, G^+ is any object containing a total match of both the object G and the left-hand side L. Therefore it can be rewritten as in the standard DPO approach, producing the two squares in the upper row. The pushout in the lower row gives the context F which we need to add to G in order to obtain G^+, along with a morphism $J \rightarrowtail F$ indicating how F should be pasted to G. Finally, the interface for the resulting object H is any object K such that the lower right-hand square commutes. If the two pushout complements, which are needed in Definition 8, namely C and F, do not exist, the rewriting step is not feasible.

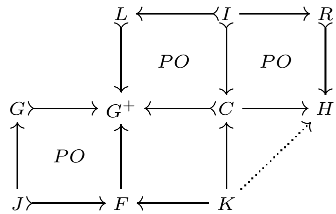


Fig. 10. A saturated derivation

Note that in this case the object G^+ might not be the minimal object containing both G and L: it could also contain something else. Therefore, the context F needed to extend G to G^+ might be not the minimal context allowing the DPO derivation for G. Fig. 11 shows an example of saturated rewriting for the graphical encoding of $\tau.\mathbf{0}$ with respect to the set of names $\Gamma = \{a\}$. Note that $-|\overline{a}$ is not the minimal context allowing the reaction: the parallel component $\overline{a}$ would not be necessary since it does non concur to the transition.

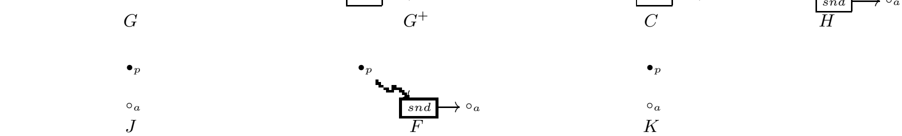


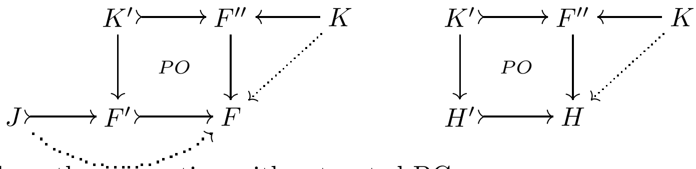
Fig. 11. The saturated transition $\tau.\mathbf{0} \xrightarrow{-|\overline{a}}_{SAT} \mathbf{0}|\overline{a}$.

Given an ARS S, we denote by $\textsc{sbc}(S)$ the LTS derived via the saturated BC mechanism as defined above. Mirroring the correspondence between BCs and GIPOs, the proposition below states that this LTS coincides with the saturated LTS for the GRS $\mathbb{C}_S$ associated to S.

Proposition 3. $\textsc{sbc}(S) = \textsc{slts}(\mathbb{C}_S)$.

Semi-Saturated Borrowed Contexts LTS. Adopting the semi-saturation point of view, it is possible to show that saturated rewriting steps can be deduced in a uniform fashion from standard rewriting steps with borrowed context.

Definition 9. *Let S be an ARS. The* semi-saturated *LTS* $\textsc{ssbc}(S)$ *is defined as the set of transitions* $J \to G \xrightarrow{J \rightarrowtail F \leftarrow K}_{SSAT} K \to H$ *such that there are a BC transition* $J \to G \xrightarrow{J \rightarrowtail F' \leftarrow K'}_{BC} K' \to H'$ *and a graphical context* $K' \rightarrowtail F'' \leftarrow K$ *making the diagrams below commute and two squares pushouts (PO).*



We now draw the connection with saturated BCs.

Proposition 4. $\textsc{sbc}(S) = \textsc{ssbc}(S)$.

Consider again the saturated transition introduced in Fig. 11. It can be derived from the minimal transition depicted in Fig. 13. More precisely, it is obtained as described in Definition 9, by considering the graphical context $K' \rightarrowtail F'' \leftarrow K$ represented in Fig. 12.

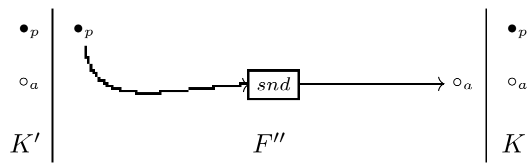


Fig. 12. The graphical context $- \mid \overline{a}$

We can exploit the above observation in order to show that the graphical encoding of $a.\overline{a} + \tau.\mathbf{0}$ and of $\tau.\mathbf{0}$ are saturated bisimilar. Consider $\sim^S$ as a semi-saturated bisimulation. If $a.\overline{a} + \tau.\mathbf{0} \xrightarrow{-|\overline{a}}_{GIPO} \overline{a} \mid \mathbf{0}$ (Fig. 16, corresponding to transition $a.\overline{a} + \tau.\mathbf{0} \xrightarrow{a} \overline{a}$), then $\tau.\mathbf{0} \xrightarrow{-}_{GIPO} \mathbf{0}$ (Fig. 13): from the latter (by Definition 9) $\tau.\mathbf{0} \xrightarrow{-|\overline{a}}_{SAT} \mathbf{0} \mid \overline{a}$ is derived (Fig. 11). Note that this is analogous to the one of asynchronous bisimulation (Definition 7). If $p \xrightarrow{a} p'$ (corresponding to $\xrightarrow{-|\overline{a}}_{GIPO}$) then $q \xrightarrow{\tau} q'$ (corresponding to $\xrightarrow{-}_{GIPO}$) with $p' R q' \mid \overline{a}$.

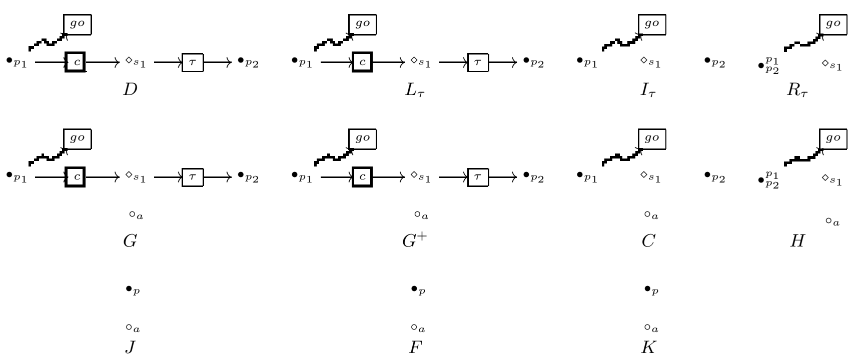


Fig. 13. The BC transition corresponding to $\tau.\mathbf{0} \xrightarrow{-}_{GIPO} \mathbf{0}$

However, $\sim^A$ is strictly included into $\sim^S$. In order to provide the simplest example possible, take into account the full process syntax, including recursion. Then, consider the processes $p = rec_X.(\tau.X + a.X)$ and $q = rec_X.\tau.X$: the latter just offers continuously an unobservable action τ, while the former may at any time perform either τ or a. Note that p is not asynchronous bisimilar to $q \mid \overline{a}$: then the same follows for p and q, since q can not simulate the reduction $p \xrightarrow{a} p$. Instead, p and $q \mid \overline{a}$ are saturated bisimilar: indeed, $\overline{a}$ remains idle in the latter process, and no parallel composition $- \mid r$ may distinguish between p and q. As it is standard for process calculi, this kind of problem can be solved by considering *barbs*, adding in the bisimulation game the check of suitable predicates over the states of a system. Next section introduces the notion of barbs for ARS.

5 Barbs for Adhesive Rewriting Systems

We have shown that neither $\sim^G$ nor $\sim^S$ capture asynchronous bisimilarity for ACCS: the former is too strict, while the latter is too coarse. The same problem arises for many formalisms (as discussed in [4]), so that the coincidence of standard bisimilarity for the synchronous CCS with $\sim^G$ appears the exception [5]. This fact lead the first three authors to introduce *barbed saturated bisimilarity* for GRSs [6]: it refines $\sim^S$ by adding state observations, called *barbs* [24]. In this section we fix a set of barbs O and write $A \downarrow_o$ if $o \in O$ holds in A.

Definition 10 (Barbed Saturated Bisimulation). *A symmetric relation $\mathcal{R}$ is a* barbed saturated bisimulation *if whenever $A \,\mathcal{R}\, B$ then*

- *$\forall C$, $A;C \,\mathcal{R}\, B;C$,*
- *if $A \downarrow_o$ then $B \downarrow_o$,*
- *if $A \rightsquigarrow A'$ then $B \rightsquigarrow B'$ and $A' \,\mathcal{R}\, B'$.*

Barbed saturated bisimilarity $\sim^{BS}$ *is the largest barbed saturated bisimulation.*

The above definition is general enough to capture the abstract semantics of many important formalisms such as mobile ambients, CCS, π-calculus and their asynchronous variants [6]. However, it is parametric with respect to the choice of the set of barbs O and defining the "right" barbs is not a trivial task, as witnessed by several papers about this topic (e.g. [29,20]).

We now introduce a novel notion of barb for ARSs, trying to keep in line with the constructive nature of the BC mechanism. To this end, our intuition is driven by the graphical encodings of calculi, and by the nature of barbs in most examples from that setting, there basically (a) barbs check the presence of some suitable subsystem, such that (b) this subsystem is needed to perform an interaction with the environment. For instance, in ACCS, barbs are parallel outputs [1], formally (a) $p \downarrow_{\overline{a}}$ if $p \equiv p_1 \mid \overline{a}$ and (b) these outputs can interact with the environment through the rule $\overline{a} \mid a.q + m \rightsquigarrow q$. In mobile ambients, barbs are ambients at the topmost level [23], formally (a) $p \downarrow_m$ if $p \equiv m[p_1] \mid p_2$ and (b) these ambients can be interact with the environment via the rule $open\, m.q_1 \mid m[q_2] \rightsquigarrow q_1 \mid q_2$.

Concerning ARSs, in order to respect intuition (a), we think of a barb D for a system G as a sub-object $D \rightarrowtail G$. Then, we assume that what can be observed on G depends on the interface J and thus we require barbs to be parametric with respect to the interfaces. Thus, a barb D for the interface J is a span $b : J \leftarrowtail J_D \rightarrowtail D$. In order to respect intuition (b), we further require that the barbs occurs in the left-hand side of some production. Formally, that D is a sub-object of L (there exists $D \rightarrowtail L$) for some production $L \leftarrowtail I \rightarrow R$.

Definition 11 (Barbs for Borrowed Contexts). *Given a production $p : L \leftarrowtail I \rightarrow R$, a system with interface $J \rightarrow G$ and spans of monos $d : G \leftarrowtail D \rightarrowtail L$ and $b : J \leftarrowtail J_D \rightarrowtail D$, we say that $J \rightarrow G$* satisfies the barb b via p *and d if there is an object F_D and additional morphisms such that the diagram in Fig. 14 commutes and one square is a pushout (PO). We write $(J \rightarrow G) \downarrow_b$.*

Put differently, a barb $J \leftarrowtail J_D \rightarrowtail D$ is satisfied by a system with interface $J \rightarrowtail G$ if it at first ensures that a suitable BC derivation can potentially be performed, even if it might not be feasible (see Definition 6 and Fig. 3). Additionally, the interface J_D is required to involve all the components of D shared with other components occurring in L (the pushout condition), while the factorisation through J guarantees that these components are uniquely identified in G.

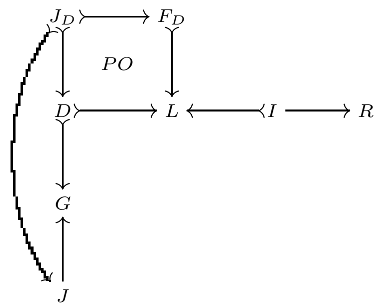


Fig. 14. The barb $J \leftarrowtail J_D \rightarrowtail D$

Consider again our running example: in ACCS we have that $\forall a \in \mathcal{N}$ we state $p \downarrow \bar{a}$ if $p \equiv \overline{a} \mid q$. The same requirement can be enforced in the graphical encoding of a process with the barb $\bar{a}_{J_D}$ shown in Fig. 15, with respect to the interface $J_D = \bullet_p \quad \circ_a$. For any larger interface J (i.e., such that there is $J \rightarrowtail J'$), the barb $\bar{a}_J$ is the obvious span $J \leftarrowtail J_D \rightarrowtail D$.

Note that the occurrence of the name node in J_D guarantees that we are observing an output on a non-restricted channel. Instead, the *go* edge in the graph D guarantees that the output observed is at the top level of the process. These output barbs are precisely those that are needed to capture $\sim^A$ via $\sim^{BS}$.

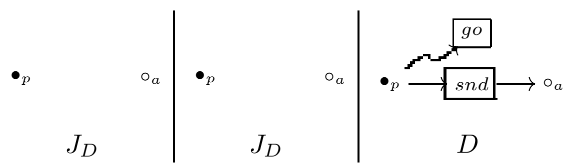


Fig. 15. Graphical barb for ACCS

Proposition 5. *Let p and q be two* ACCS *processes. Let Γ be a set of names such that $fn(p) \cup fn(q) \subseteq \Gamma$. Let $[\![p]\!]_\Gamma$ and $[\![q]\!]_\Gamma$ be the graphical encodings of p and q with respect to Γ as described in Section 3, and let the barbs $\bar{a}_J$ be as defined above. Then $p \sim^A q$ iff $[\![p]\!]_\Gamma \sim^{BS} [\![q]\!]_\Gamma$.*

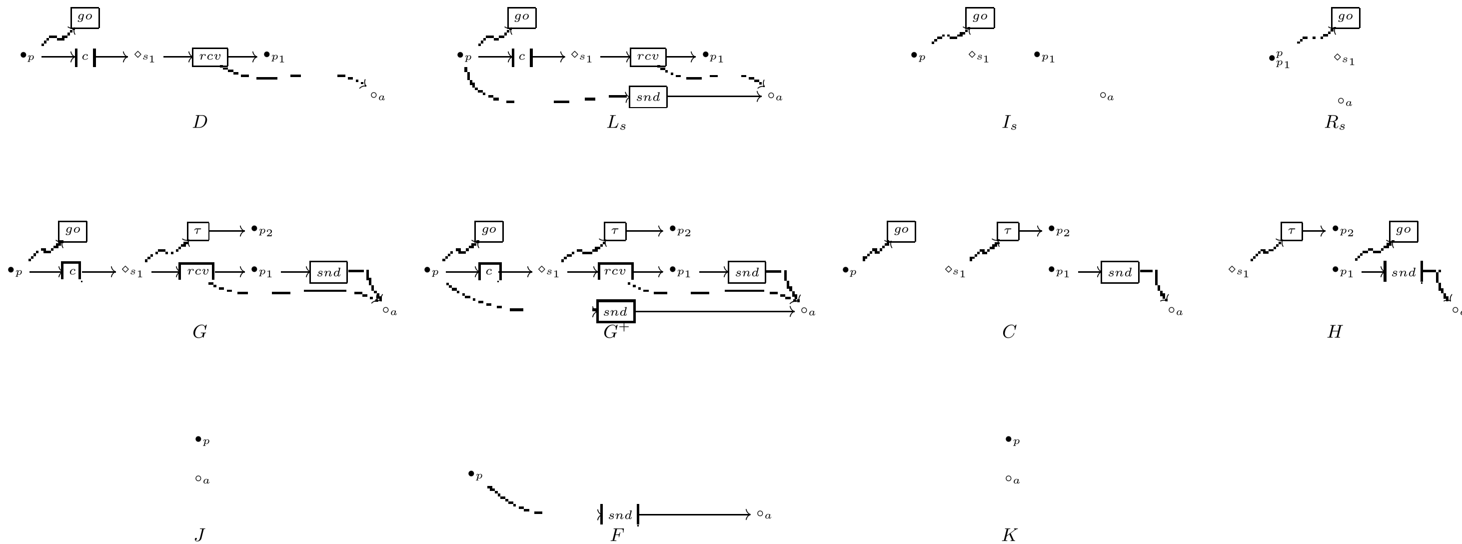


Fig. 16. The BC transition corresponding to the GIPO transition $a.\overline{a} + \tau.\mathbf{0} \xrightarrow{-|\overline{a}}_{GIPO} \overline{a}|\mathbf{0}$

6 Conclusions and Further Works

The aim of the paper is to add flexibility to the BC mechanism. To this end, we introduced saturated LTSs for ARSs, proving their correspondence to a constructive presentation of the analogous notion for GRSs. We also presented barbed semantics for ARSs: they are needed to address the semantics of many process calculi, such as asynchronous CCS and mobile ambients, but their application could be wider. For instance, [28] studies model refactoring that preserves GIPO bisimilarity: the latter could be safely replaced by barbed saturated bisimilarity.

Our work also opens new venues of investigation. Indeed, barbs in the GRS setting are quite general, hence quite poor from e.g. the point of view of its modularity properties. On the contrary, the simpler notion of barbs that we devised for BCs, besides its constructive features, may represent a major step towards solving the problem of automatically deriving suitable barbs for GRSs.

Finally, we believe that the modelling of calculi is one of the strongest assets of the BC mechanism: graphical encodings for processes can be easily provided, and their reduction semantics simulated by DPO rules. By integrating our results with *negative application conditions* [19] (adapted to the BC framework in [27]) we might provide suitable behavioural theories for larger classes of calculi.

References

1. Amadio, R., Castellani, I., Sangiorgi, D.: On bisimulations for the asynchronous π-calculus. Theoretical Computer Science 195(2), 291–324 (1998)
2. Bénabou, J.: Introduction to bicategories. In: Midwest Category Seminar I. LNM, vol. 47, pp. 1–77. Springer, Heidelberg (1967)
3. Birkedal, L., Debois, S., Hildebrandt, T.: On the construction of sorted reactive systems. In: van Breugel, F., Chechik, M. (eds.) CONCUR 2008. LNCS, vol. 5201, pp. 218–232. Springer, Heidelberg (2008)
4. Bonchi, F.: Abstract Semantics by Observable Contexts. Ph.D. thesis, Department of Informatics, University of Pisa (2008)
5. Bonchi, F., Gadducci, F., König, B.: Synthesising CCS bisimulation using graph rewriting. Information and Computation 207(1), 14–40 (2009)
6. Bonchi, F., Gadducci, F., Monreale, G.V.: Labelled transitions for mobile ambients (as synthesized via a graphical encoding). In: Hildebrandt, T., Gorla, D. (eds.) EXPRESS 2008. ENTCS, vol. 242(1), pp. 73–98. Elsevier, Amsterdam (2009)
7. Bonchi, F., Gadducci, F., Monreale, G.V.: Reactive systems, barbed semantics, and the mobile ambients. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 272–287. Springer, Heidelberg (2009)
8. Bonchi, F., König, B., Montanari, U.: Saturated semantics for reactive systems. In: LICS 2006, pp. 69–80. IEEE Computer Society, Los Alamitos (2006)
9. Bonchi, F., Brogi, A., Corfini, S., Gadducci, F.: On the use of behavioural equivalences for web services' development. Fundamenta Informaticae 89(4), 479–510 (2008)
10. Cardelli, L., Gordon, A.: Mobile ambients. Theoretical Computer Science 240(1), 177–213 (2000)
11. Cardelli, L.: Brane calculi. In: Danos, V., Schachter, V. (eds.) CMSB 2004. LNCS (LNBI), vol. 3082, pp. 257–278. Springer, Heidelberg (2005)
12. Ehrig, H., König, B.: Deriving bisimulation congruences in the DPO approach to graph rewriting with borrowed contexts. Mathematical Structures in Computer Science 16(6), 1133–1163 (2006)

13. Gadducci, F.: Graph rewriting for the π-calculus. Mathematical Structures in Computer Science 17(3), 407–437 (2007)
14. Gadducci, F., Heckel, R.: An inductive view of graph transformation. In: Parisi-Presicce, F. (ed.) WADT 1997. LNCS, vol. 1376, pp. 219–233. Springer, Heidelberg (1998)
15. Gadducci, F., Montanari, U.: A concurrent graph semantics for mobile ambients. In: Brookes, S., Mislove, M. (eds.) MFPS 2001. ENTCS, vol. 45. Elsevier, Amsterdam (2001)
16. Gianantonio, P.D., Honsell, F., Lenisa, M.: RPO, second-order contexts, and lambda-calculus. Logical Methods in Computer Science 5(3) (2009)
17. Grohmann, D., Miculan, M.: Reactive systems over directed bigraphs. In: Caires, L., Vasconcelos, V.T. (eds.) CONCUR 2007. LNCS, vol. 4703, pp. 380–394. Springer, Heidelberg (2007)
18. Grohmann, D., Miculan, M.: Deriving barbed bisimulations for bigraphical reactive systems. In: Corradini, A., Tuosto, E. (eds.) ICGT 2008 - Doctoral Symposium. Electronic Communications of the EASST, vol. 16. EASST (2009)
19. Habel, A., Heckel, R., Taentzer, G.: Graph grammars with negative application conditions. Fundamenta Informaticae 26(3/4), 287–313 (1996)
20. Honda, K., Yoshida, N.: On reduction-based process semantics. Theoretical Computer Science 151(2), 437–486 (1995)
21. Lack, S., Sobocinski, P.: Adhesive and quasiadhesive categories. Theoretical Informatics and Applications 39(3), 511–545 (2005)
22. Leifer, J., Milner, R.: Deriving bisimulation congruences for reactive systems. In: Palamidessi, C. (ed.) CONCUR 2000. LNCS, vol. 1877, pp. 243–258. Springer, Heidelberg (2000)
23. Merro, M., Zappa Nardelli, F.: Behavioral theory for mobile ambients. Journal of the ACM 52(6), 961–1023 (2005)
24. Milner, R., Sangiorgi, D.: Barbed bisimulation. In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 685–695. Springer, Heidelberg (1992)
25. Milner, R.: Bigraphical reactive systems. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 16–35. Springer, Heidelberg (2001)
26. OWL-S Coalition: OWL-S for services, `www.ai.sri.com/daml/services/owl-s/`
27. Rangel, G., König, B., Ehrig, H.: Deriving bisimulation congruences in the presence of negative application conditions. In: Amadio, R.M. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 413–427. Springer, Heidelberg (2008)
28. Rangel, G., Lambers, L., König, B., Ehrig, H., Baldan, P.: Behavior preservation in model refactoring using DPO transformations with borrowed contexts. In: Ehrig, H., Heckel, R., Rozenberg, G., Taentzer, G. (eds.) ICGT 2008. LNCS, vol. 5214, pp. 242–256. Springer, Heidelberg (2008)
29. Rathke, J., Sassone, V., Sobociński, P.: Semantic barbs and biorthogonality. In: Seidl, H. (ed.) FOSSACS 2007. LNCS, vol. 4423, pp. 302–316. Springer, Heidelberg (2007)
30. Rathke, J., Sobociński, P.: Deriving structural labelled transitions for mobile ambients. In: van Breugel, F., Chechik, M. (eds.) CONCUR 2008. LNCS, vol. 5201, pp. 462–476. Springer, Heidelberg (2008)
31. Sassone, V., Sobocinski, P.: Deriving bisimulation congruences using 2-categories. Nordic Journal of Computing 10(2), 163–183 (2003)
32. Sassone, V., Sobociński, P.: Reactive systems over cospans. In: LICS 2005, pp. 311–320. IEEE Computer Society Press, Los Alamitos (2005)
33. Sobociński, P.: Deriving bisimulation congruences from reduction systems. Ph.D. thesis, BRICS, Department of Computer Science, University of Aaurhus (2004)