

A Modular LTS for Open Reactive Systems

Fabio Gadducci, Giacomina Monreale, Ugo Montanari

► **To cite this version:**

Fabio Gadducci, Giacomina Monreale, Ugo Montanari. A Modular LTS for Open Reactive Systems. 7th International Conference on Theoretical Computer Science (TCS), Sep 2012, Amsterdam, Netherlands. pp.134-148, 10.1007/978-3-642-33475-7_10 . hal-01556218

HAL Id: hal-01556218

<https://hal.inria.fr/hal-01556218>

Submitted on 4 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Modular LTS for Open Reactive Systems^{*}

Fabio Gadducci, Giacomina Valentina Monreale and Ugo Montanari

Dipartimento di Informatica, Università di Pisa, Italy

Abstract. The theory of reactive systems (RSs) represents a fruitful proposal for deriving labelled transition systems (LTSs) from unlabelled ones. The synthesis of an LTS allows for the use of standard techniques in the analysis of systems, as witnessed by the widespread adoption of behavioral semantics. Recent proposals addressed one of the main drawbacks of RSs, namely, its restriction to the analysis of ground (i.e., completely specified) systems. A still unresolved issue concerns the lack of a presentation via inference rules for the derived LTS, thus hindering the modularity of the presentation. Our paper considers open RSs. We first introduce a variant of the current proposal based on “luxes”: our technique is applicable to a larger number of case studies and, under some conditions, it synthesises a smaller LTS. Then, we illustrate how the LTS derived by using our approach can be equipped with a SOS-like presentation via an encoding into tile systems.

Keywords: Open reactive systems, labelled transition systems, tile systems.

1 Introduction

The ever increasing diffusion of concurrent and distributed systems stimulated the development of novel formalisms for their specification. These formalisms usually provide an abstract presentation of the behaviour of such a system by resorting to some kind of operational description, possibly adopting also an observational equivalence over the system configurations.

At its simplest, the dynamics of a computational model is defined by means of a *reduction semantics*: a set representing the possible states of the system, plus an unlabelled relation among these states, denoting the potential evolutions of the system. The set of states is often provided by means of an equational specification, referred to as “structural congruence” in the process calculi literature, stating which presentations intuitively specify the same system, up to a syntactical rearrangement of its components.

Despite the advantage of conveying the semantics with relatively few rules, the main drawback of reduction semantics is that the dynamics of a system is described in a monolithic way. Thus, it can be observationally interpreted only by inserting a system in appropriate contexts, where a reduction may take place.

^{*} Partly supported by the EU FP7-ICT IP ASCEns and by the MIUR PRIN SisteR.

To ease the analysis of systems, it is often necessary to consider descriptions allowing the analysis of the behaviour of each single subcomponent, thus increasing modularity and enhancing the opportunities for verification. In such a context, labelled transition systems (LTSs) represent the most widely used tool. LTSs open the way for the definition of observational equivalences, abstractly characterising when two systems have the same behaviour, thus allowing the possibility of verifying the properties of system composition. However, the identification of the “right” labels is a difficult task and it is usually left to the ingenuity of the researcher. A case at hand is the Calculus of Mobile Ambients (MA) [5]: despite its rapid acceptance and the intense scrutiny it was subject to, the development of a suitable labelled semantics defied the researchers until quite recently [11], and chances are that it might not be fully settled [1,15].

The theory of *reactive systems* (RSs) [9] represents one of the most successful meta-frameworks among those addressing the need of deriving suitable LTSs and behavioural equivalences starting from reduction semantics. The key idea is simple: a system h has a labelled transition $h \xrightarrow{c} h'$ if the system obtained by inserting h inside the “minimal” context c may reduce to h' . The framework uses categories to model the state space of a formalism and it exploits the categorical notion of relative pushout (RPO) to capture the intuitive notion of “minimal” environment into which a system specification has to be inserted, in order to allow for a reduction to occur. However, in several examples where calculi with even simple structural congruences are considered, the theory cannot be immediately applied. So, A futher advance is represented by G-reactive systems (GRSs), [16], a 2-categorical extension of RSs that allows for an easier representation of those calculi such that the structural congruence is an integral part of the theory.

A success story for GRSs is represented by the application to MA, and the proposal of slender LTSs, yet inducing the same behavioural semantics of the original proposal [11] (see [1,15]). Despite its applicability, the main limit suffered by GRSs is the restriction to the use of ground rules for describing the dynamics of a system: often a strong requirement, in the modelling of open-ended systems operating in an ever changing environment. Consider e.g. Milner’s CCS, and the reduction rule modelling the communication over a channel a , namely $a.P \mid a.Q \rightarrow P \mid Q$. In the GRS framework, it is represented by an infinte set of ground rules, one for each possible pair P and Q of processes. One would instead like to have a more general and succinct theory allowing to express parametric rules such as $a.1 \mid a.2 \rightarrow 1 \mid 2$, for 1 and 2 placeholders that can be freely instantiated.

This paper aims at addressing this problem, by developing a theory considering also open terms and parametric rewriting rules. So now the transitions will be labelled not only with the minimal context but also with the most general instantiation allowing a reduction. More explicitly, $h \xrightarrow[c]{x} h'$ if h instantiated with the (possibly open) term x and inserted into the context c may evolve into a state h' . Quite intuitively, our proposal exploits the notions of RPO and the symmetric relative pullback (RPB), suitably extended to GRSs, to capture the notions of minimal context and most general instantiation, respectively.

The framework recalls the one in [8] based on *luxes*, the only approach we are aware of for an open variant of GRSs. Their formalism advances a notion of labelled transition that captures at once the requirements on context and substitution. Our proposal weakens the constraints and it can thus be applied to more case studies, as well as generating, under some conditions, a smaller LTS.

Another clear limitation of the standard GRSs approach is represented by the lack of a finitary presentation of the derived LTS, possibly via a set of inference rules according to the SOS-style. So, following the solution for closed GRSs suggested in [3], we draw the connection with the tile model [6], showing how to build a (double) category of “squares” that generates precisely the LTS derived by an open GRS according to the proposal offered in this paper.

The main concepts of the paper are illustrated by a small running example, based on a Simplified Calculus of Mobile Ambients (SMA). Despite its minimal syntax, with respect to more basic, CCS-like calculi, SMA will allow us to better show the usefulness of taking into account contexts and instantiations.

The paper is organized as follows. Sect. 2 briefly recalls 2-categories. Sect. 3 introduces open GRSs and the technique that we propose to derive LTSs, while Sect. 4 presents a preliminary comparison with the theory in [8]. Sect. 5 shows how to construct a tile system simulating the reductions of an open GRS. Sect. 6 concludes the paper by illustrating some venues for further works.

2 Some background on 2-Categories

This section introduces the 2-categorical definitions used later on (see [4,7]).

Definition 1 (2-category). *A 2-category \mathbf{C} consists of*

1. *a family of objects a, b, c, \dots ;*
2. *for each $a, b \in \mathbf{C}$ a category $\mathbf{C}(a, b)$. The objects of $\mathbf{C}(a, b)$ are called 1-cells or arrows and denoted by $f : a \rightarrow b$. Identity arrows are instead denoted by $id_a : a \rightarrow a$. The morphisms are called 2-cells, and are written $\alpha : f \Rightarrow g : a \rightarrow b$. Composition in $\mathbf{C}(a, b)$ is denoted by \bullet and referred to as vertical composition. Identity 2-cells are denoted by $1_f : f \Rightarrow f$;*
3. *for each $a, b, c \in \mathbf{C}$ a functor $*$: $\mathbf{C}(a, b) \times \mathbf{C}(b, c) \rightarrow \mathbf{C}(a, c)$, called horizontal composition. It is associative and admits 1_{id_a} as identities.*

The functoriality of $*$ amounts to imposing the *exchange law* $(\alpha \bullet \gamma) * (\beta \bullet \delta) = (\alpha * \beta) \bullet (\gamma * \delta)$ for any 4-tuple α, β, γ and δ of composable cells.

2-categories are used to simulate the reduction semantics of formalisms [14,12], where reductions are modelled by 2-cells. Starting from an abstract presentation of the basic reduction steps of a system, the closure with respect to contexts is then obtained by the 2-categorical operation of whiskering [7].

Definition 2 (G-category). *A groupoidal category (*G-category*) is a 2-category whose 2-cells are invertible.*

We will show an example of G-category based on the PROP category [10].

Definition 3. A product and permutation (PROP) category \mathbf{C} has natural numbers $0, 1, 2, \dots$ as objects and it is equipped with two further structures

- for each n , the group of permutations $S(n)$ of n elements is a subgroup of all the invertible elements of the homset $\mathbf{C}(n, n)$. The identity permutation is the identity morphism $1_n : n \rightarrow n$;
- a functor $\otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$, called product and written between its arguments, which acts as addition on the objects, i.e., $m \otimes n = m + n$, and such that
 1. it is associative, i.e., $(f \otimes f') \otimes f'' = f \otimes (f' \otimes f'')$;
 2. given $\sigma \in S(n)$ and $\sigma' \in S(n')$, we have $\sigma \otimes \sigma' = \sigma \times \sigma' : n + n' \rightarrow n + n'$, where \times denotes the product of permutations;
 3. given $f : m \rightarrow n$ and $f' : m' \rightarrow n'$, we have that $\gamma_{n, n'}(f \otimes f') = (f' \otimes f)\gamma_{m, m'}$, where $\gamma_{n, n'} : n + n' \rightarrow n + n'$ denotes the permutation in $S(n + n')$ interchanging the first block of n and the second block of n' .

A G-PROP is a PROP where the underlying category is a G-category.

Example 1 (GPROP PA2CP). We now show how a signature of a calculus (modulo term equations) may induce a G-PROP. Inspired by the characterization of (a fragment of) CCS in [8], we model a simplified version of Mobile Ambients (MA). The richer structure of this calculus, as well as the complexity of its basic reduction rules, will later allow us to highlight the need of considering open terms and parametric reduction rules.

Let us then denote SMA the finite, restriction free fragment of MA with only the *in* action. The syntax of the calculus is shown on the left of the upper row of Fig. 1. We assume a set \mathcal{N} of *names* ranged over by m, n, u, \dots . We let P, Q, R, \dots range over the set \mathcal{P} of closed processes, containing no holes.

The semantics is given by the combination of an equivalence between processes, the structural congruence, and a relation among them. The former, denoted by \equiv , is induced by the two rightmost axioms in the first row of Fig. 1. It is used to define the reduction relation \rightsquigarrow , which is inductively generated by the axioms and inference rules shown in the lower rows of Fig. 1. It is used only later on, but we introduce it here for the sake of presentation.

$$\begin{array}{l}
 P ::= \epsilon, n[P], in\ n.P, P_1|P_2 \qquad (P|Q)|R \equiv P|(Q|R) \qquad P|Q \equiv Q|P \\
 \hline
 n[in\ m.P|Q]|m[R] \rightsquigarrow m[n[P|Q]|R] \qquad \text{if } P \rightsquigarrow Q \text{ then } P|R \rightsquigarrow Q|R \\
 \text{if } P \rightsquigarrow Q \text{ then } n[P] \rightsquigarrow n[Q] \qquad \text{if } P' \equiv P, P \rightsquigarrow Q, Q \equiv Q' \text{ then } P' \rightsquigarrow Q'
 \end{array}$$

Fig. 1. Syntax, structural congruence and reduction relation of SMA.

The signature corresponding to SMA is $\Sigma = \epsilon : 0, n[] : 1, in\ n. : 1, | : 2$.

The G-PROP PA2CP has arrows $h : m \rightarrow n$, representing n -tuples of terms over Σ , quotiented by associativity, that altogether contain m distinct holes. Permutations in (n, n) are tuples built solely of holes, \otimes acts on arrows as tuple juxtaposition, and arrow composition is term substitution.

To define the 2-cells of the category, an explicit representation of the arrows of the category is used. A term can be indeed represented as a finite, ordered tree with nodes of any degree, where an immediate child of a node of degree higher than 1 must have degree at most 1. Leaves of such a tree correspond to occurrences of the constant ϵ . Nodes of degree 1 correspond to an application either of an ambient operator or of a capability. In the former case the node is labelled with a name belonging to \mathcal{N} , while in the latter case it is labelled with the capability followed by an ambient name. Finally, nodes of higher degree correspond to term fragments built solely of the parallel operator. So, arrows can be represented as tuples of these trees.

A 2-cell from h to h' models the equivalence of h and h' according to the commutative axiom. So, it is a family, indexed by the nodes of (the explicit representation of) h , of permutations on the sets of their immediate children, such that the application of all these permutations to h yields h' .

3 A New LTS for Open Reactive Systems

This section presents an extension of *G-reactive systems* (GRSs) [16]. The theory aims at deriving labelled transition systems (LTSs) for specification formalisms whose operational semantics is provided by reduction rules. The technique was originally given for closed systems, that is, closed terms and ground reduction rules. As in [8], our proposal considers terms with variables and parametric rules. The idea is simple: a system specified by an open term h has a labelled transition $h \xrightarrow[c]{x} h'$ if h may evolve into a state h' after being instantiated with the (possibly open) term x and inserted into the context c .

A G -category \mathbf{C} models the syntax of a formalism. An (*open*) *system* is an arrow $h : a_1 \rightarrow a_2$: it can be plugged into $g : a_2 \rightarrow a_3$ via arrow composition. Given arrows $h, g : a_1 \rightarrow a_2$, a 2-cell $\alpha : h \Rightarrow g$ represents an isomorphism (i.e., a proof of equivalence) between systems h and g . The semantics is given via *reduction rules*: pairs of systems $\langle l, r \rangle$ belonging to the same hom-set.

Definition 4 (Open GRS). *An open G-reactive system (GRS) \mathbb{C} consists of*

1. a G -category \mathbf{C} ;
2. a composition-reflecting, 2-cell closed, subcategory \mathbf{D} of reactive contexts;
3. a set $\mathfrak{R} \subseteq \bigcup_{a_1, a_2 \in |\mathbf{C}|} \mathbf{C}(a_1, a_2) \times \mathbf{C}(a_1, a_2)$ of reduction rules.

Intuitively, reactive contexts are those arrows inside which a reduction can occur. By 2-cell closed we mean that $d \in \mathbf{D}$ and $\alpha : d \Rightarrow d'$ in \mathbf{C} implies $d' \in \mathbf{D}$, while by composition-reflecting we mean that $d'; d \in \mathbf{D}$ implies $d, d' \in \mathbf{D}$.

Given an open GRS \mathbb{C} , the reduction relation over the terms of \mathbb{C} is generated by closing the reduction rules under all reactive contexts, instantiations and 2-cells. Formally, the *reduction relation* is defined by taking $h \rightsquigarrow h'$ if there exist $\langle l, r \rangle \in \mathfrak{R}$, $d \in \mathbf{D}$, $x \in \mathbf{C}$, $\alpha : h \Rightarrow x; l; d$ and $\alpha' : h' \Rightarrow x; r; d$.

Example 2. Consider the G-category G-PROP PA2CP shown in Example 1. We can then construct an open GRS \mathbb{C}_{SMA} over it by taking as the set of reduction rules the set $\bigcup_{n,m \in \mathcal{N}} \{\langle n[in\ m.1 \mid 2] \mid m[3], m[n[1 \mid 2] \mid 3] \rangle\}$ (denoting here and in the following, with an abuse of notation, id_i with i for $i = 1, 2, 3$), and as the subcategory of reactive contexts the smallest composition-reflecting, 2-cell closed subcategory including arrows of the shape $1 \mid P : 1 \rightarrow 1$ and $n[1] : 1 \rightarrow 1$. Note that it does not contain contexts with a hole after a capability.

The behaviour of an open GRS is given by an unlabelled transition system. To obtain a labelled one, we instantiate an open system h with a subterm x , plug the result into a context c and observe if a reduction occurs. Categorically, it means that $x; h; c$ is isomorphic to $y; l; d$ (there exists $\alpha : x; h; c \Rightarrow y; l; d$) for an instantiation x , a rule $\langle l, r \rangle$, and a reactive context d (Fig. 2(a)).

The resulting LTS is often infinite-branching, since any context (or instantiation) allowing a reduction may occur as label. It also has redundant transitions: the SMA open term $T = n[in\ m.1 \mid 2]$ would have both transitions $T \xrightarrow[\langle P, Q \rangle]{1[m[R]]} m[n[P \mid Q] \mid R]$ and $T \xrightarrow[\langle P, Q \rangle]{1[m[R]]S} m[n[P \mid Q] \mid R] \mid S$, yet S does not “concur” to the reduction. We thus consider only “minimal contexts allowing a reduction”, captured by the categorical notion of groupoidal-idem pushouts (GIPOs) in G-categories, and the “most general instantiations”, modelled by the categorical notion of groupoidal-idem pullbacks (GIPBs). We refer to [16] for the notion of GIPO, the one of GIPB being perfectly symmetric.

Definition 5 (GIPO-GIPB LTS). Let \mathbb{C} be an open GRS and \mathbf{C} its underlying G-category. The GIPO-GIPB LTS $\text{GGLTS}(\mathbb{C})$ is defined as follows

- states: $h : a_1 \rightarrow a_2$ in \mathbf{C} , for arbitrary a_1 and a_2 ;
- transitions: $h \xrightarrow[x]{c} h'$ if there exist $d \in \mathbf{D}$ and $\langle l, r \rangle \in \mathfrak{R}$ such that the upper rhombus of the diagram in the middle of Fig. 2 is a GIPO, the lower one is a GIPB, and $h' = y; r; d$.

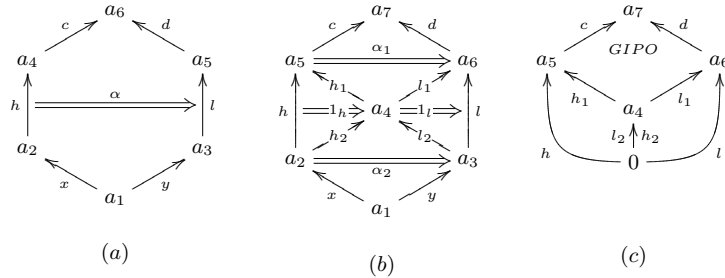


Fig. 2. The 2-cell $\alpha : x; h; c \Rightarrow y; l; d$, a labelled transition $h \xrightarrow[x]{c}$ and a GIPO.

Example 3. Consider the open GRS \mathbb{C}_{SMA} introduced in Example 2. The left-most diagram of Fig. 3 shows the derived transition $\langle n[in\,m.1 \mid Q], 3 \rangle \xrightarrow{1|m[3]} m[n[1 \mid Q] \mid 3]$ (omitting here and in the following any identity instantiation such as $\langle 1, 3 \rangle$ in a label). The initial state offers an ambient containing a capability $in\,m$ in parallel with a closed process Q , while the environment provides the ambient m which is going to be in parallel with n , so the reduction can occur. Only the identity instantiation is instead provided.

The transition $\langle n[1 \mid Q], 3 \rangle \xrightarrow[\langle in\,m.1, 3 \rangle]{1|m[3]} m[n[1 \mid Q] \mid 3]$ is shown in the diagram on the right of Fig. 3: it needs both a context and an instantiation. The initial open term again offers an ambient n , but now it contains a hole in parallel with Q . Therefore, the instantiation provides the capability $in\,m.1$, replacing the hole inside n , while the environment once more provides the ambient m .

This transition shows the importance of using both instantiations and contexts. In this case, we can note the interaction of the three parts occurring in the reduction: the initial state, the instantiation and the context. This situation never occurs in simpler calculi such as CCS, where we can only have the synchronization between two systems in parallel.

Remark 1. The original notion of closed GRS [16] is very similar to the one for open GRSs. One difference is that in the category \mathbf{C} there exists a chosen object 0 , which is the codomain of no arrow and denotes the lack of holes. Arrows having 0 as domain are deemed to represent closed terms. Another difference is that the rules are closed, i.e., they are pairs of closed terms, and also the states of the derived (GIPO) LTS are so: $h \xrightarrow{c} h'$ if there exist $d \in \mathbf{D}$ and $\langle l, r \rangle \in \mathfrak{R}$, such that the square h, c, l, d is a GIPO and $h' = r; d$.

Our proposal may be specialized to consider either closed terms or closed rules, and it then subsumes the standard GRS formalism. Indeed, let us assume that rules and terms are ground. This means that in the diagram (b) of Fig. 2 the objects a_1, a_2 and a_3 are equal to 0 . We therefore obtain the diagram (c) of Fig. 2, and if e.g. l_2 is an epimorphism then the external square of diagram (c) is a GIPO, and so $h \xrightarrow{c}$ in the standard GIPO LTS.

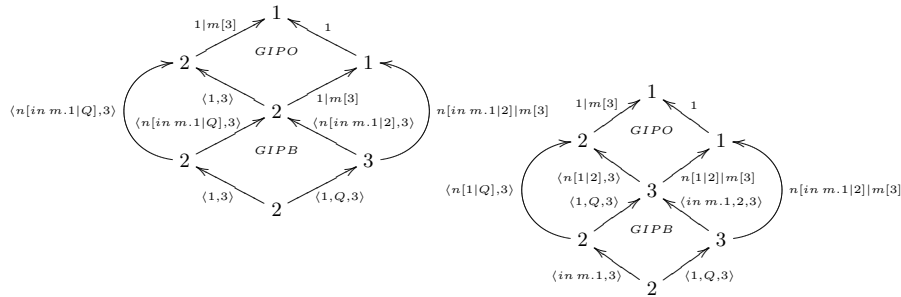


Fig. 3. Transitions $\langle n[in\,m.1 \mid Q], 3 \rangle \xrightarrow{1|m[3]} T$ and $\langle n[1 \mid Q], 3 \rangle \xrightarrow[\langle in\,m.1, 3 \rangle]{1|m[3]} T$.

4 On G-luxes

This section presents a preliminary comparison between the technique presented in the previous section and the one in [8]. These proposals share the same idea: labels should include both the minimal context and the most general instantiation allowing a reduction. As we will argue, the main conceptual difference is that in [8] the authors introduce the novel notion of *G-locally universal hexagon* (*G-lux*) in order to capture at the same time the two components of a label via a suitable universal property, while our proposal considers contexts and instantiations separately, thus resorting to standard GIPOs and GIPBs, respectively.

Our definition turns out to encompass a wider range of case studies. Indeed, categories with G-luxes always have GIPOs and GIPBs [8, Theorem 1], and from that fact our statement follows. We now recall the notion of G-lux LTS adopting a simple characterization via GIPOs and GIPBs [8, Lemma 10].

Definition 6 (G-lux). *The hexagon (a) in Fig. 2 is a G-locally universal hexagon (G-lux) if in Fig. 4, (a) is a GIPO and (b) is a GIPB.*

*A G-category has G-luxes if every hexagon as diagram (a) in Fig. 2 has an inner G-lux as diagram (c) of Fig. 4, and $\alpha = \delta * 1_h * \beta \bullet 1_z * \psi * 1_e \bullet \varphi * 1_l * \gamma$.*

Definition 7 (GLUX Transition System). *Let \mathbb{C} be an open GRS and \mathbf{C} its underlying G-category. The GLUX LTS ($\text{LLTS}(\mathbb{C})$) is defined as follows*

- states: $h : a_1 \rightarrow a_2$ in \mathbf{C} , for arbitrary a_1 and a_2 ;
- transitions: $h \xrightarrow[x]{c} h'$ if there exist $d \in \mathbf{D}$, $\langle l, r \rangle \in \mathfrak{R}$ and $\alpha : x; h; c \Rightarrow y; l; d$ such that the diagram (a) of Fig. 2 is a G-lux and $h' = y; r; d$.

The diagrams of Fig. 5 show two examples of G-luxes in the category PA2CP. The left-most one gives the labelled transition $n[in\ m.1 \mid 2] \xrightarrow[1 \mid m[R]]{m[n[1 \mid 2] \mid R]}$. This transition shows one of the main problems luxes suffer from, already discussed in the last paragraph of [8, Example 5]: the intertwining of context and instantiation may sometimes result in the offering of components that are not necessary for the reduction. In particular, in the example R is redundant, since it appears both in the context on the left and in the instantiation on the right. Moreover, since R is arbitrary, the derived LTS is infinitely branching.

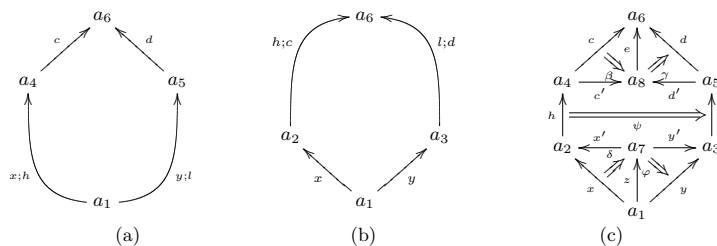


Fig. 4. G-lux.

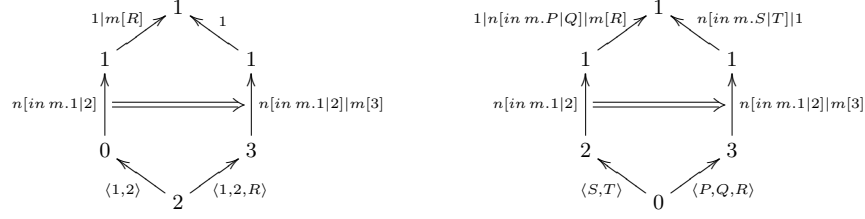


Fig. 5. G-luxes transitions originating from $n[in\ m.1 \mid 2]$.

If we apply our theory to derive the labelled transitions for the same process $n[in\ m.1 \mid 2]$, it is easy to verify that no labelled transition originates from it. For every splitting of the process and the right-hand side of the reduction rule, it is indeed never possible to obtain a GIPO-GIPB diagram.

Another example of G-lux in the category PA2CP is depicted on the right of Fig. 5. It represents the labelled transition $n[in\ m.1 \mid 2] \xrightarrow[\langle S,T \rangle]{1 \mid n[in\ m.P|Q] \mid m[R]}$

$n[in\ m.S \mid T] \mid m[n[P \mid Q] \mid R]$, where the context offers all the components that are needed to the reduction. Moreover, there is once more redundancy: processes S and T appear both in the instantiation on the left and in the context on the right, and processes P , Q and R appear both in the instantiation on the right and in the context on the left. Instead, by applying our theory we cannot derive this transition, and indeed, it is not an interesting transition: it is not relevant for the bisimilarity, since it is possible to derive it for any term.

From the examples above, it seems that our theory performs better at obtaining LTSs with less redundancy. This can be stated formally under certain conditions, as shown by the theorem below.

Definition 8. Let \mathbf{C} be a G -category, $f_1, f_2 : a_1 \rightarrow a_2$ two morphisms of \mathbf{C} and $\alpha : f_1 \Rightarrow f_2$ a 2-cell. We say that it is a right-factor if for all $g_1, g_2 : a_0 \rightarrow a_1$ and $\beta : g_1 ; f_1 \Rightarrow g_2 ; f_2$ there exists $\gamma : g_1 \Rightarrow g_2$ such that $\gamma * \alpha = \beta$.

Let $f : a_1 \rightarrow a_2$ be a morphism of \mathbf{C} . We say that it is a 2-monomorphism if for all $g_1, g_2 : a_0 \rightarrow a_1$ and $\gamma, \gamma' : g_1 \Rightarrow g_2$, $\gamma * 1_f = \gamma' * 1_f$ implies $\gamma = \gamma'$.

The dual notions of right-factor and 2-mono are respectively the one of left-factor and 2-epi, respectively, with 2-iso defined as obvious.

Theorem 1. Let \mathbb{C} be an open GRS such that its underlying category has G-luxes and all its arrows are 2-iso. If $h \xrightarrow[x]{c} h'$ belongs to $\text{GGLTS}(\mathbb{C})$ (Fig. 2 (b)), α_1 is a right- and α_2 a left-factor, then $h \xrightarrow[x]{c} h'$ also belongs to $\text{LLTS}(\mathbb{C})$.

5 From Open G-Reactive Systems to Tile Systems

In this section we show how an open GRS can be used to generate a 2-category: this fact is going to be used to associate to a GRS a tile system, hence to obtain a finitary presentation of the derived LTS via a set of inference rules.

From Open GRSs to 2-Categories. In order to derive a 2-category from an open GRS, we adapt the construction used in [3] for RSs with closed rules.

Definition 9 (2-category of interactions). *Let \mathbf{C} be an open GRS $\langle \mathbf{C}, \mathbf{D}, \mathfrak{R} \rangle$. Then, \mathbf{C}_i denotes the 2-category freely generated from $\langle \mathbf{C}, \mathfrak{R} \rangle$.*

The 2-cells in \mathbf{C}_i are freely generated from the G-category \mathbf{C} and the reduction rules \mathfrak{R} , by adding the identity cells and closing under vertical and horizontal composition, subject to the exchange law. A 2-cell of \mathbf{C}_i may not denote a meaningful computation in \mathbf{C} , since also reductions inside non-reactive contexts are allowed. Differently from [3], it indeed seems impossible to identify a suitable sub-2-category of \mathbf{C}_i precisely characterizing the arrows of the open LTS: the problem is that reductions can occur only inside open terms representing reactive contexts, while any open term can represent a possible instantiation. We therefore need a way to distinguish the two types of arrows, and to this end, it is necessary to resort to *double categories*.

From 2-Categories to Double Categories. We now recall a construction for deriving a double category of squares from a 2-category [13]. As shown later, it suggests an automatic generation of a labelled relation (abstracted by a double category) starting from an unlabelled one (abstracted by a 2-category).

In the following, we fix a chosen 2-category \mathbf{C} .

Definition 10 (C squares). *Let a, a', b, b' objects and $h : a \rightarrow a', g : b \rightarrow b'$ 1-cells of \mathbf{C} . The four sets of \mathbf{C} squares, each square characterized by a 6-tuple (signature; left, right; top, bottom; inside) of data from \mathbf{C} , are defined below.*

Square $S_{-/-}$ from 1-cells $v_- : a \rightarrow b$ and $u_- : a' \rightarrow b'$ and 2-cell $\alpha_{-/-} : h; u_- \Rightarrow v_-; g$ (first diagram in Fig. 6). We indicate the data for $S_{-/-}$ by the 6-tuple $d(S_{-/-}) = (-/-; v_-, u_-; h, g; \alpha_{-/-})$ of data from \mathbf{C} . The left identity $l(S_{-/-})$ for $S_{-/-}$ is $(v_-, v_-; id_a, id_b; 1_{v_-})$, the right one is $r(S_{-/-}) = (u_-, u_-; id_{a'}, id_{b'}; 1_{u_-})$, the top one is $t(S_{-/-}) = (id_a, id_{a'}; h, h; 1_h)$ and the bottom one is $b(S_{-/-}) = (id_b, id_{b'}; g, g; 1_g)$.

Square $S_{+/+}$ from 1-cells $v_+ : a \leftarrow b$ and $u_+ : a' \leftarrow b'$ and 2-cell $\alpha_{+/+} : v_+; h \Rightarrow g; u_+$ (second diagram in Fig. 6), with data $(+/+; v_+, u_+; h, g, \alpha_{+/+})$ and $l(S_{+/+}) = (v_+, v_+; id_a, id_b; 1_{v_+})$, $r(S_{+/+}) = (u_+, u_+; id_{a'}, id_{b'}; 1_{u_+})$, $t(S_{+/+}) = (id_a, id_{a'}; h, h; 1_h)$, and $b(S_{+/+}) = (id_b, id_{b'}; g, g; 1_g)$.

Square $S_{+/-}$ from 1-cells $v_+ : a \leftarrow b$ and $u_- : a' \rightarrow b'$ and 2-cell $\alpha_{+/-} : v_+; h; u_- \Rightarrow g$ (third diagram in Fig. 6), with data $(+/-; v_+, u_-; h, g, \alpha_{+/-})$ and $l(S_{+/-}) = (v_+, v_+; id_a, id_b; 1_{v_+})$, $r(S_{+/-}) = (u_-, u_-; id_{a'}, id_{b'}; 1_{u_-})$, $t(S_{+/-}) = (id_a, id_{a'}; h, h; 1_h)$, and $b(S_{+/-}) = (id_b, id_{b'}; g, g; 1_g)$.

Square $S_{-/+}$ from 1-cells $v_- : a \rightarrow b$ and $u_+ : a' \leftarrow b'$ and 2-cell $\alpha_{-/+} : v_-; g; u_+ \Leftarrow h$ (last diagram in Fig. 6), with data $(-/+; v_-, u_+; h, g; \alpha_{-/+})$ and $l(S_{-/+}) = (v_-, v_-; id_a, id_b; 1_{v_-})$, $r(S_{-/+}) = (u_+, u_+; id_{a'}, id_{b'}; 1_{u_+})$, $t(S_{-/+}) = (id_a, id_{a'}; h, h; 1_h)$, and $b(S_{-/+}) = (id_b, id_{b'}; g, g; 1_g)$.

In the following we define the vertical and the horizontal composition.

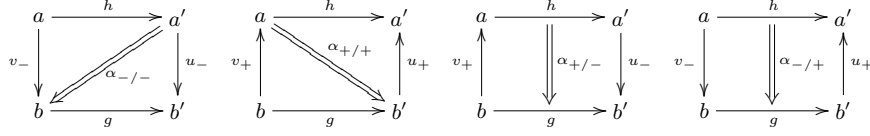


Fig. 6. C squares.

Definition 11 (Vertical Composition). Let S and S' be two C squares with arbitrary signatures a/b and c/d , respectively, and such that $b(S)$ equals $t(S')$. This forces the equality of signatures $a = c$ and $b = d$. So, the data of S and S' have respectively the form $(a/b; v, u; h, g; \alpha)$ and $(a/b; v', u'; g, g'; \alpha')$, while the one for the composition $S \bullet S'$ are $(a/b; v'', u''; h, g'; \alpha'')$. Items v'', u'' and α'' vary with the signature. Since there are four possibilities for original signatures (both a and b can have $+$ and $-$ as possible values) there are four cases of vertical composition listed in the leftmost table of Fig. 7.

C squares form a category $(\mathbf{C Sq})^\bullet$ under \bullet composition.

a/b	v''	u''	α''
$+/+$	$v'; v$	$u'; u$	$(1_{v'} * \alpha) \bullet (\alpha' * 1_u)$
$+/-$	$v'; v$	$u; u'$	$(1_{v'} * \alpha * 1_{u'}) \bullet \alpha'$
$-/+$	$v; v'$	$u'; u$	$\alpha \bullet (1_v * \alpha' * 1_u)$
$-/-$	$v; v'$	$u; u'$	$(\alpha * 1_{u'}) \bullet (1_v * \alpha')$

a/b	α''
$(+, +, +)$	$(\alpha * 1_{h'}) \bullet (1_g * \alpha')$
$(+, +, -)$	$(\alpha * 1_{h'} * 1_{u'}) \bullet (1_g * \alpha')$
$(+, -, +)$	$(1_v * 1_h * \alpha') \bullet (\alpha * 1_{g'} * 1_{u'})$
$(+, -, -)$	$(1_v * 1_h * \alpha') \bullet (\alpha * 1_{g'})$
$(-, +, +)$	$(\alpha * 1_{h'}) \bullet (1_v * 1_g * \alpha')$
$(-, +, -)$	$(\alpha * 1_{h'} * 1_{u'}) \bullet (1_v * 1_g * \alpha')$
$(-, -, +)$	$(1_h * \alpha') \bullet (\alpha * 1_{g'} * 1_{u'})$
$(-, -, -)$	$(1_h * \alpha') \bullet (\alpha * 1_{g'})$

Vertical composition

Horizontal Composition

Fig. 7. Instances of vertical and horizontal compositions.

Note that in both tables the vertical and horizontal compositions used in the last columns are the ones of the 2-category C.

Definition 12 (Horizontal Composition). Let S and S' be two C squares with arbitrary signatures a/b and c/d , respectively, and such that $r(S)$ equals $l(S')$. This forces the equality of signatures $b = c$. So, the data of S and S' have respectively the form $(a/b; v, u; h, g; \alpha)$ and $(b/d; u, u'; h', g'; \alpha')$, while the one for the composition $S * S'$ are $(a/d; v, u'; h; h', g; g'; \alpha'')$. Item α'' varies with the signature. Since there are eight possibilities for original signatures (each of a , b and d can have $+$ and $-$ as possible values) there are eight cases of horizontal composition, which are listed in the rightmost table of Fig. 7.

C squares form a category $(\mathbf{C Sq})^*$ under $*$ composition.

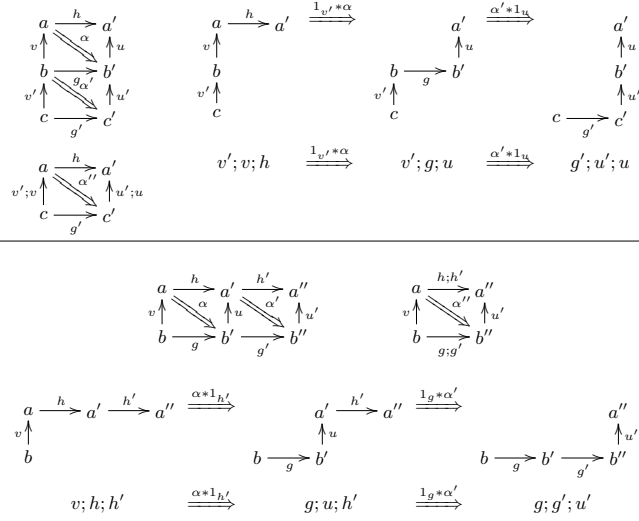


Fig. 8. Vertical and horizontal composition of two $+/\mathbf{C}$ squares.

The upper part of Fig. 8 depicts the first case of vertical composition of the leftmost table of Fig. 7. The leftmost diagram on the top shows the two squares to be composed, the leftmost diagram on the bottom depicts the resulting square, and the remaining part shows how the 2-cell α'' is obtained.

The lower part of Fig. 8 depicts an example of horizontal composition. Again, we consider two \mathbf{C} squares with signature $+/\mathbf{C}$. So, the leftmost diagram on the top shows the two squares to be composed, the diagram on the right depicts the resulting square, and the remaining part shows how the 2-cell α'' is obtained.

Proposition 1. *Squares and compositions above form a double category $\mathbf{C}Sq$, for $(\mathbf{C}Sq)^*$ and $(\mathbf{C}Sq)^\bullet$ the horizontal and vertical category, respectively.*

From GRSs to double categories. Some cells of $\mathbf{C}Sq$ do not represent labelled transitions derived by the GIPO-GIPB mechanism. We thus introduce a way to cut them, precisely characterising the LTS synthesised from an open GRS.

Definition 13 (Observational double category). *Let $\mathbb{C} = \langle \mathbf{C}, \mathbf{D}, \mathfrak{R} \rangle$ be an open GRS. The observational double category of \mathbb{C} , denoted $\mathcal{O}(\mathbb{C})$, is the smallest sub-double category of $\mathbf{C}_i Sq$ which includes the cells in Fig. 9 where the cells of type (i) correspond to GIPBs in \mathbf{C} , those of type (ii) to the rules of \mathfrak{R} , with $l_2; l_1 = l$ for l the left-hand side of a rule in \mathfrak{R} , and those of type (iii) to GIPOs in \mathbf{C} , with $g \in \mathbf{D}$.*

The squares of $\mathcal{O}(\mathbb{C})$ with signature $+/-$ correspond to the derivations of $\text{GGLTS}(\mathbb{C})$, as stated by the theorem below.

Theorem 2. *Let \mathbb{C} be an open GRS. The $\text{GGLTS}(\mathbb{C})$ has a transition $h \xrightarrow[c]{c} h'$ if and only if in $\mathcal{O}(\mathbb{C})$ there is a \mathbf{C} square with data $(+/-; x, c; h, h'; \alpha)$.*

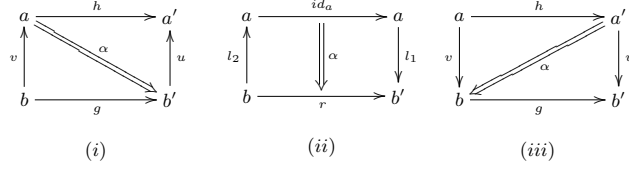


Fig. 9. The basic cells generating the observational double category of \mathbb{C} .

Example 4. Consider the open GRS $\mathbb{C}_{SMA} = \langle \mathbf{C}, \mathbf{D}, \mathfrak{R} \rangle$ introduced in Example 2. Its observational double category $\mathcal{O}(\mathbb{C}_{SMA})$ is the smallest sub-double category of the double category $\mathbf{C}_i Sq$ which includes the cells of type (i), (ii) and (iii), as stated in the previous definition.

In particular, it contains the cells corresponding to the only rule of \mathfrak{R} , some of which are shown in Fig. 10. Among the cells of type (i) and (iii) there are the ones in the upper row of Fig. 11, since they are respectively two GIPBs and two GIPOs, and moreover the context offering just a hole 1 is reactive. Therefore, it is easy to see that from the composition of the above cells we obtain the cells in the upper row of Fig. 12, corresponding to the labelled transitions in Fig. 3.

Also the lower cell represents a labelled transition. It can be obtained from the one above it by composing this with the leftmost and rightmost squares of the lower transition, which are respectively a GIPB and a GIPO. It is easy to see that it represents the transition $\langle n[1 \mid Q] \mid S, 3 \mid R \rangle \xrightarrow{\langle in\ m.1,3 \rangle} m[n[1 \mid Q] \mid 3 \mid R] \mid S$. Indeed, the cell in the middle of the diagram is of type (ii), and since the two rightmost cells are GIPOs, it is obvious that also the cell obtained composing them is so; similarly, the cell obtained composing the two leftmost squares is a GIPB, since the two squares are so.

Remark 2. Let us now consider a closed GRS. Then, for any cell of type (ii) the object b always coincide with the chosen object 0. In order to allow horizontal composition, also for any cell of type (i) both b and b' should always coincide with 0 and, should we consider only closed terms, also a would be 0. In the generation of the observation double category it would then suffice to consider as cells of type (i) only those \mathbf{C} squares with data $(+ / +; id_0, l_2; l_2, id_0; \alpha)$, for l_2 an epimorphism. We thus recover the observational double category for ground rules and terms proposed in [3, Definition 11].

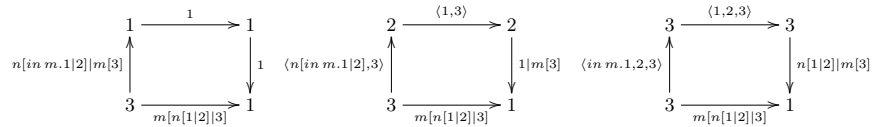


Fig. 10. Some cells corresponding to the rule $\langle n[in\ m.1 \mid 2] \mid m[3], m[n[1 \mid 2] \mid 3] \rangle$.

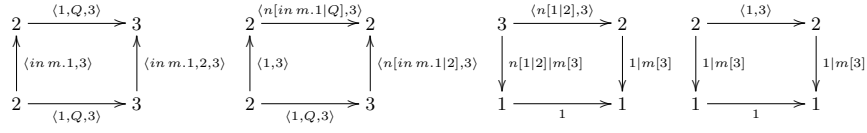


Fig. 11. Cells of $\mathcal{O}(C_{SMA})$ of type (i) and (iii).

6 Conclusions and Further Works

We presented a mechanism to derive LTSs for GRSs with open terms and parametric rules. With respect to the only alternative proposal we are aware of [8], our approach almost straightforwardly exploits the well-known categorical notions of GIPO and GIPB to respectively capture the notions of minimal context and most general instantiation that allow for a reduction to occur.

We illustrated the main concepts by using a small running example, a simplified version of Mobile Ambients. The use of SMA allowed us to better highlight the role of instantiations and contexts in labelled transitions for open terms. It also gave the chance of performing an informal comparison between our approach and the one of [8]. This is made precise by observing that our approach can be used for a wider range of case studies and that it seems to derive in general a more concise LTS (even if this has been formally proved so far only under quite stringent conditions). However, we leave as future work the comparison of the results obtained by applying the two techniques to real-life formalisms, as well as the comparison between our technique and the more complex synthesis mechanism based on irredundant G-luxes (also championed in [8]).

As in [3], our paper draws a connection with the tile model [6]: the derivation from an open GRS \mathbb{C} of a (freely constructed) double category is illustrated, exploiting a functorial construction proposed in [13]. It results in a finitary, SOS-like presentation of the LTS derived from that same \mathbb{C} via a suitable double category. Preliminary observations suggest that the LTS synthesised via G-luxes cannot be offered the same modular presentation: the simultaneous requirement on contexts and instantiations seems to require an infinite set of basic tiles in the observational double category, in order to model G-lux transitions via cells.

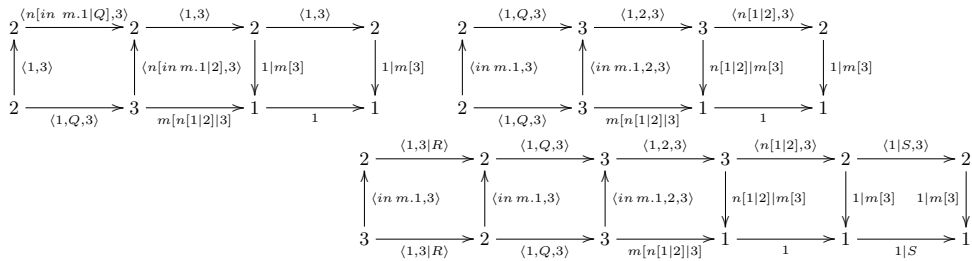


Fig. 12. Cells of $\mathcal{O}(C_{SMA})$ representing labelled transitions.

Despite its features, our framework suffers from the same problem as the synthesis mechanism based on irredundant G-luxes: the bisimilarity over the derived LTS is not a congruence. Consider e.g. the SMA process $n[in\ m.1 \mid 2]$. As said in Section 4, no labelled transition starting from it can be derived. This implies that it is equivalent to $n[in\ o.1 \mid 2]$, for any channel name o . Clearly, this equivalence is not preserved by all contexts. We leave as future work the study of this problem. It could lead towards a further refinement of our synthesis mechanism for LTSs, possibly by acting directly on the generating cells of the observational double category. Otherwise, one could consider an alternative definition of bisimilarity: as it has been carried out for the closed GRSs approach [2], one could study barbed saturated bisimilarity for open terms, by giving a labelled characterization of it through the GGLTS.

References

1. Bonchi, F., Gadducci, F., Monreale, G.V.: Labelled transitions for mobile ambients (as synthesized via a graphical encoding). In: EXPRESS 2008. ENTCS, vol. 242(1), pp. 73–98. Elsevier (2009)
2. Bonchi, F., Gadducci, F., Monreale, G.V.: Reactive systems, barbed semantics, and the mobile ambients. In: FoSSaCS 2009. LNCS, vol. 5504, pp. 272–287. Springer (2009)
3. Bruni, B., Gadducci, F., Montanari, U., Sobocinski, P.: Deriving weak bisimulation congruences from reduction systems. In: CONCUR 2005. LNCS, vol. 3653, pp. 293–307. Springer (2005)
4. Bruni, R., Meseguer, J., Montanari, U.: Symmetric monoidal and cartesian double categories as a semantics framework for tile logic. MSCS 12(1), 53–90 (2002)
5. Cardelli, L., Gordon, A.: Mobile ambients. TCS 240(1), 177–213 (2000)
6. Gadducci, F., Montanari, U.: The tile model. In: Proof, Language and Interaction: Essays in Honour of Robin Milner, pp. 133–166. MIT Press (2000)
7. Kelly, G., Street, R.: Review of the elements of 2-categories. In: Sydney Category Seminar. LNM, vol. 420, pp. 75–103. Springer (1974)
8. Klin, B., Sassone, V., Sobocinski, P.: Labels from reductions: Towards a general theory. In: CALCO 2005. LNCS, vol. 3629, pp. 30–50. Springer (2005)
9. Leifer, J., Milner, R.: Deriving bisimulation congruences for reactive systems. In: CONCUR 2000. LNCS, vol. 1877, pp. 243–258. Springer (2000)
10. MacLane, S.: Categorical algebra. Bull. Amer. Math. Soc. 71, 40–106 (1965)
11. Merro, M., Zappa Nardelli, F.: Behavioral theory for mobile ambients. Journal of the ACM 52(6), 961–1023 (2005)
12. Meseguer, J.: Conditional rewriting logic as a unified model of concurrency. TCS 96(1), 73–155 (1992)
13. Palmquist, P.: The double category of adjoint squares. In: Midwest Category Seminar. LNM, vol. 195, pp. 123–153. Springer (1971)
14. Power, A.: An abstract formulation for rewrite systems. In: CTCS 1989. LNCS, vol. 389, pp. 300–312. Springer (1989)
15. Rathke, J., Sobociński, P.: Deriving structural labelled transitions for mobile ambients. In: CONCUR 2008. LNCS, vol. 5201, pp. 462–476. Springer (2008)
16. Sassone, V., Sobocinski, P.: Deriving bisimulation congruences using 2-categories. Nordic Journal of Computing 10(2), 163–183 (2003)