Deriving weak bisimulation congruences from reduction systems*

Roberto Bruni, Fabio Gadducci, Ugo Montanari, and Paweł Sobociński

Dipartimento di Informatica, Università di Pisa, Italia.

Abstract. The focus of process calculi is *interaction* rather than *computation*, and for this very reason: (i) their operational semantics is conveniently expressed by labelled transition systems (LTSs) whose labels model the possible interactions of a process with the environment; (ii) their abstract semantics is conveniently expressed by observational congruences. However, many current-day process calculi are more easily equipped with reduction semantics, where the notion of *observable action* is missing. Recent techniques attempted to bridge this gap by synthesising LTSs whose labels are process contexts that enable reactions and for which bisimulation is a congruence. Starting from Sewell's set-theoretic construction, category-theoretic techniques were defined and based on Leifer and Milner's *relative pushouts*, later refined by Sassone and the fourth author to deal with structural congruences given as *groupoidal 2-categories*.

Building on recent works concerning observational equivalences for *tile logic*, the paper demonstrates that *double categories* provide an elegant setting in which the aforementioned contributions can be studied. Moreover, the formalism allows for a straightforward and natural definition of weak observational congruence.

1 Introduction

Since Milner's proposal of an alternative semantics for the π-calculus [14] based on reactive rules modulo a suitable structural congruence, ongoing research focused on the investigation of the relationship between the *labelled transition system* (LTS) based semantics for process calculi and the more abstract *reduction semantics*.

Early attempts by Sewell [19] devised a strategy for obtaining an LTS from a *reduction relation* by adding suitable contexts as labels on transitions. The technique was further refined by Leifer and Milner [11] who introduced the notion of *relative pushout* (RPO) in order to capture the notion of *minimal context*. Such attempts share the basic property of synthesising a congruent bisimulation equivalence.

In this paper we pursue the comparison between these two different semantic styles, using categorical tools to model and to relate the possible approaches. The result is a schema for the translation of reductions semantics into LTS semantics such that their natural bisimulation equivalences are indeed congruences with respect to the state structure. In particular, we show that double categories provide a uniform framework for experimenting with different constructions of observational models out of reactive systems, accounting for both weak and strong bisimulation congruences.

* This work has been partly supported by the EU within the project HPRN-CT-2002-00275 SegraVis (*Syntactic and Semantic Integration of Visual Modelling Techniques*).

$$\begin{tikzcd} 0 \arrow[r, "P"] & 1 \arrow[r, bend left, "\alpha.[\_]|\bar{\alpha}"] \arrow[r, bend right, "{[\_]}"'] \arrow[r, phantom, "\tau\Downarrow"] & 1 \arrow[r, "C[\_]"] & 1 \end{tikzcd}$$

Fig. 1. The reduction $C[\alpha.P|\bar{\alpha}] \Rightarrow C[P]$.

Reduction semantics. The dynamics of many calculi is often defined in terms of reduction relations. For example, the λ-calculus has the β-reduction $(\lambda x.M)N \Rightarrow M[N/x]$ that models the application of a functional process $\lambda x.M$ to the actual argument N. Usually, this kind of rules can be freely instantiated and contextualised because they represent *internal reductions* of a system component. For example the reduction rule $\alpha.P|\bar{\alpha} \Rightarrow P$ for asynchronous CCS-like communication can be instantiated to $P = \bar{\beta}$ and contextualised in the unary context $C[\_] = \beta.nil|[\_]$ yielding the rewrite sequent

$$\beta.nil|\alpha.\bar{\beta}|\bar{\alpha} \Rightarrow \beta.nil|\bar{\beta}$$

illustrated in Fig. 1 with a standard notation: natural numbers represent the number of context-holes, hence an arrow from 0 to 1 is a ground process, while an arrow from 1 to 1 is a context with a unique hole. Processes and contexts compose horizontally, while reductions proceed vertically.

Observational semantics. Reduction semantics have the advantage of conveying the semantics of calculi with relatively few compact rules. The main drawback of reduction semantics is poor compositionality, in the sense that the dynamic behaviour of arbitrary stand alone terms (like $\alpha.P$ in the example above) can be interpreted only by inserting them in the appropriate context (i.e., $[\_]|\bar{\alpha}$), where a reduction may take place. Instead, in LTS semantics, transitions are labelled over suitable *observable actions*; these are intended to capture the potential interactions of each process with any environment. Because interaction is explicit, this approach has proven to be flexible in defining various notions of process equivalence.

Reductions vs. labelled transitions, or cells vs. double cells. Both reductions and labeled transitions have a strong set-theoretic flavour. Nevertheless, both logical and categorical presentations for these paradigms have been proposed in the literature. Concerning reduction semantics, it is agreed that *enriched categories* (more specifically, *2-categories*) are a suitable model [16, 17], and *rewriting logic* [13] a successful logical framework for interpreting many computational formalisms. Concerning LTSs, *tile logic* [8] offers a uniform approach to system specifications, admitting both a sequent calculus presentation (with rules accounting for side-effects and synchronisations) and a categorical semantics in terms of *double categories*. Moreover, tile logic yields a natural notion of observational equivalence, *tile bisimulation*, for which congruence proofs can be carried out in a purely diagrammatic way. Our belief is that the comparison between reduction semantics and LTS semantics can be conveniently pursued at the level of their categorical representatives.

Indeed, Leifer and Milner's notion of reactive system can be seen as a 2-category in which the 2-cells are freely generated from a set of basic ground *reaction rules*. This treatment generalises to Sassone and the fourth author's work, where the starting point is a special kind of 2-category that accounts for structural congruences, called a G-category, and adds such reductions freely to obtain a 2-category.

In order to study the derivations on a LTS, we construct the *observational double category* out of a reduction system, which expresses the orthogonality of reactions (the vertical dimension of the double category) and contexts (the horizontal dimension). This double category unites all of the structure (terms, contexts, structural congruence and reductions) in the same categorical universe, and it allows us to recover Leifer and Milner's notion of strong bisimilarity. More interestingly, the ordinary notion of tile bisimilarity turns out to define a congruent *weak* bisimilarity which promises to be an operationally more natural equivalence.

Structure of the paper. In Section 2 we recall the definitions of double categories, 2-categories and tile bisimulation. In Section 3 we recall the definition of reactive system and show how the theory can be reconciled with traditional 2-categorical approaches to rewriting. Section 4 is devoted to the main contribution of the paper, showing that: (1) depth preserving tile bisimulation over observational double categories is a congruence which corresponds to Leifer and Milner's strong bisimilarity, and (2) ordinary tile bisimilarity results in a notion of weak bisimulation congruence. In the Conclusion we summarise the results and point out further extensions and other possible applications of our framework. In the Appendix we give some technical background on previous work relating to the notions of reactive systems and relative pushouts.

2 Background

Double categories. This section presents a minimal introduction to double categories; we refer the reader to [2, 8] for further details. Throughout the paper we shall follow the convention of denoting composition in the diagrammatic order.

Concisely, a *double category* is simply an internal category in Cat (the category of small categories and functors). This means that a double category contains two categorical structures, called *horizontal* and *vertical* respectively, defined over the same set of cells. More explicitly, double categories admit the following, naïve definition.

Definition 1 (Double category). *A* double category $\mathcal{D}$ *consists of a collection of* cells $\alpha, \beta, \gamma, ...$ *such that*

1. *cells form the* horizontal category $\mathcal{D}^*$*, where* $*$ *denotes horizontal cell composition;*
2. *cells form the* vertical category $\mathcal{D}^\bullet$*, where* $\bullet$ *denotes vertical cell composition;*
3. *the objects of* $\mathcal{D}^*$*, ranged by* $v, u, w, ...$*, are called* observations *and form the* vertical 1-category $\mathcal{V}$ *over the objects in* O*, ranged by* $a, b, c, ...$*;*
4. *the objects of* $\mathcal{D}^\bullet$*, ranged by* $h, g, f, ...$*, are called* configurations *and form the* horizontal 1-category $\mathcal{H}$ *over the same objects* O *of* $\mathcal{V}$*;*
5. *both vertical and horizontal composition of cells are* functorial *with each other and w.r.t. the corresponding compositions in the underlying 1-categories.*

$$\begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "\alpha"] & b \arrow[d, "u"] \\ c \arrow[r, "g"'] & d \end{tikzcd}$$

Fig. 2. Graphical representation of a cell.

We shall often use ';' to denote composition in both the horizontal and vertical 1-categories. A cell α with horizontal source v, horizontal target u, vertical source h and vertical target g is written $\alpha : h \xrightarrow[u]{v} g$ and depicted as in Fig. 2—its sources and targets must be *compatible*, in the sense that h and v must have the same domain a, the codomain of v must coincide with the domain of g and so on, as illustrated in Fig. 2.

The functoriality requirement amounts to impose the convenient *exchange law*

$$(\alpha \bullet \gamma) * (\beta \bullet \delta) = (\alpha * \beta) \bullet (\gamma * \delta)$$

for any composable cells $\alpha : h \xrightarrow[u]{v} g$, $\beta : f \xrightarrow[w]{u} l$, $\gamma : g \xrightarrow[x]{z} h'$, and $\delta : l \xrightarrow[y]{x} f'$.

To substantiate the definition of a double category, we give some basic examples.

Example 1 (Square category). Given a category C, the corresponding *double category of squares* is defined by taking the objects of C as objects, C as both the horizontal 1-category and vertical 1-category, and the set of *square diagrams* formed by compatible arrows (in the sense explained above) as cells.

Example 2 (Quartet category). Cells of a square category are compatible, but not necessarily commuting. Given a category C, we denote by $\Box\, C$ the *double category of quartets* of C: its objects are the objects of C, its horizontal and vertical arrows are the arrows of C, and its cells are the *commuting* square diagrams of arrows in C (i.e., such that $h; u = v; g$ with reference to Fig. 2). The quartet category is therefore a sub-double category of the square category.

Since any square in the square category over C is uniquely characterised by its "border" (i.e., any two squares with the same border are equal), it is immediate that in all the examples above the exchange law is trivially satisfied. Note that in general a double category can have many different cells with the same border. Indeed, when considering double categories which arise from 2-categories using a generalisation of the quartet construction, we shall consider two cells to be equal if they have equal border *and* equal internal 2-cells (Definition 10).

2-categories. A *2-category* is described concisely as a double category whose underlying vertical 1-category is discrete (i.e., it only contains identity arrows). In other terms a *2-category* C is a category where every homset (the collections of arrows between any pair of objects a and b) is the class of objects of some category $C(a, b)$ and, correspondingly, whose composition "functions" $C(a, b) \times C(b, c) \to C(a, c)$ are functors.

Definition 2 (2-category). *A 2-category C consists of*

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

Definition 3 (G-category). *A* groupoidal category *(or G-category) is a 2-category where all 2-cells are invertible.*

Starting from [16, 17, 13], 2-categories have been the chosen formalism for the algebraic presentation of the reduction semantics for many term-like structures [5, 7]—the 2-cells of such 2-categories model reduction. The idea is to start from an abstract presentation of the basic reduction steps of a system: the closure with respect to contexts is then precisely obtained by the 2-categorical operation of whiskering [20]. Here, the relevant notion is that of *(G-)computad*: a (G-)category enriched with a relation on homsets, each pair representing a basic reduction step of the system. Via a well-known construction, a 2-category can be freely generated from any (G-)computad.

Definition 4 (G-computad). *A* G-computad *is a pair $\langle \mathcal{H}, T \rangle$, where $\mathcal{H}$ is a G-category and $T = \bigcup_{a,b \in \mathcal{H}} T_{a,b}$ is a family of relations on arrows $T_{a,b} \subseteq \mathcal{H}(a, b) \times \mathcal{H}(a, b)$.*

When writing $f\ T\ g$ we assume that f, g belong to the same homset. G-computads are slightly more general than computads, in that $\mathcal{H}$ is a G-category instead of an ordinary category (which itself can be seen as a G-category whose 2-cells are all identities).

Ground tile bisimilarity. When used as a semantic foundation for computational models as tile logic, double categories allow for a suitable notion of behavioural equivalence which is reminiscent of the well-known technique of *bisimulation*. This notion can be lifted to a more abstract level of generic double categories without much effort.

The general definition of tile bisimulation establishes a family of equivalences for each homset in the horizontal category $\mathcal{D}^*$. A restricted variant, called *ground tile bisimulation* in [4], focuses just on the suitable homset of *closed* processes; it is relevant for us because reactive systems (in the sense of Leifer and Milner, see Definition 7) are designed for closed systems. In our framework, closed systems correspond to horizontal arrows which cannot be left-instantiated, except in trivial ways.

In the following we shall assume that our horizontal 1-category has a distinguished *ground object* ι: we require that for all objects a, if there exists $f : a \to \iota$ then $f = id_\iota$. The closed systems we shall consider are then characterised by having a ground object in their left interface and we simply write $t \xrightarrow[v]{} t'$ for a cell with horizontal source id_ι.

Definition 5 (Ground tile bisimulation). *Let $\mathcal{D}$ be a double category with a ground object ι. A symmetric relation B on closed configurations (arrows in the homsets $\mathcal{H}[\iota, a]$ for any object a) is called a* ground tile bisimulation *if whenever $s\ B\ t$ and $s \xrightarrow[v]{} s' \in \mathcal{D}$, then t' exists such that $t \xrightarrow[v]{} t' \in \mathcal{D}$ and $s'\ B\ t'$.*

$$
\begin{tikzcd}
\iota \arrow[r, "s_1"] \arrow[d] & a_1 \arrow[r, "h"] & a \arrow[d, "v"] \\
\iota \arrow[rr, "t"'] & & b
\end{tikzcd}
\qquad
\begin{tikzcd}
\iota \arrow[r, "s_1"] \arrow[d] & a_1 \arrow[r, "h"] \arrow[d, "u"] & a \arrow[d, "v"] \\
\iota \arrow[r, "t_1"'] & b_1 \arrow[r, "g"'] & b
\end{tikzcd}
$$

Fig. 3. Ground decomposition.

The maximal ground tile bisimulation is denoted by $\approx$, and two closed configurations s and t are *ground tile bisimilar* if $s \approx t$. Note that $\approx$ only relates arrows within the same homset. Bisimilarity is said to be *congruent* when $s \approx t$ implies $s; c \approx t; c$ for any arrow c in $\mathcal{D}^*$. The following property on $\mathcal{D}$ is known to be sufficient for congruence.

Definition 6 (Ground decomposition). *A double category $\mathcal{D}$ enjoys the* ground decomposition property *if for any ground configuration $s : \iota \to a$ and any cell $s_1; h \xrightarrow[v]{} t \in \mathcal{D}$ such that $s = s_1; h$, there exist an observation u, a ground configuration t_1 and a configuration g such that $s_1 \xrightarrow[u]{} t_1 \in \mathcal{D}$ and $h \xrightarrow[v]{u} g$, with $t = t_1; g$.*

The situation is depicted in Fig. 3. The observation u defines the amount of interaction between s_1 and the environment h that is needed to perform the effect v. In general u is not uniquely determined, as s_1 and h can interact in many ways. For example, it can be that $u = id$ if h can perform v without interacting with s_1. The key point about the (ground) decomposition property is that a transition of the whole can always be expressed as a suitable combination of the transitions of its parts.

Theorem 1 (Cfr. [4]). *The ground decomposition property implies that ground tile bisimilarity is a congruence.*

3 From reactive systems to 2-categories

Reactive systems were proposed by Leifer and Milner as a general framework for the study of simple formalisms equipped with a reduction semantics [11]. The setting was extended by Sassone and the fourth author [18] in order to treat the situation where the contexts of a formalism are equipped with a structural congruence relation. For instance, in examples which contain a parallel composition operator, it is usually not satisfactory to simply quotient out terms with respect to its commutativity—intuitively, it is important to know the precise location within the term where the reaction occurs. This information is expressed in a natural way as a 2-dimensional structure, where the 2-cells are isomorphisms which "permute" the structure of the term.

Definition 7 (Reactive system). *A* reactive system $\mathbb{C}$ *consists of*

1. *a G-category C of* context*;*
2. *a distinguished object $\iota \in C$;*
3. *a composition-reflecting, 2-full 2-subcategory $\mathcal{E}$ of* evaluation contexts*;*[1]
4. *a set of pairs $R \subseteq \bigcup_{a \in \mathcal{E}} C(\iota, a) \times C(\iota, a)$ called the* reaction rules.

[1] That is, $\mathcal{E}$ is full on the two-dimensional structure and $e_1; e_2 \in \mathcal{E} \Rightarrow e_1 \in \mathcal{E}$ and $e_2 \in \mathcal{E}$.

Reaction rules are closed with respect to evaluation contexts in order to obtain the reaction relation on the closed terms (arrows with domain ι) of C.

A calculus with restriction. As a running example, we shall first define a G-category C, the arrows of which shall represent the terms of a simple process calculus with a restriction operator. Adding the expected reaction rules, we shall obtain a reactive system.

Objects. Two objects: $0, 1$.

Arrows. The homset $C(0, 0)$ is the singleton containing only the identity arrow. There are no arrows from 1 to 0. Fixing a set A of channel names, we construct the terms of our simple calculus as specified by the grammar below

$$P \quad ::= \quad \epsilon \quad | \quad a \quad | \quad \overline{a} \quad | \quad - \quad | \quad P \,\mathbf{|}\, P \quad | \quad \nu a.P \qquad (a \in A)$$

Although the parallel composition '$\mathbf{|}$' is a binary operator, we shall consider terms to be quotiented with respect to its associativity. The set of closed terms (those terms containing no occurrences of the hole '$-$') is the homset $C(0, 1)$. The set of terms which contain precisely one hole forms the homset $C(1, 1)$.

Composition of C arrows (either an arrow $t : 0 \to 1$ with an arrow $c : 1 \to 1$, or two arrows $c : 1 \to 1$ and $d : 1 \to 1$) is substitution of the first term for the unique hole within the second term. Note that the hole in an open term is allowed to be within the scope of a restriction, and thus substitution can involve capturing.

2-cells. Roughly, the structural isomorphisms between terms of our G-category C correspond to the usual axioms describing the commutativity of '$\mathbf{|}$', while at the same time respecting the scopes of any present restriction.

More concretely, 2-cells between terms without restriction are permutations which swap parallel components (where by 'component' we mean an occurrence of an input/output on a channel or a hole). Thus, for instance, there are two automorphisms on $a \,\mathbf{|}\, a : 0 \to 1$, the identity, and the automorphism which swaps the two copies of a.[2]

The restriction $\nu a.P$ reduces the allowed permutations in any context: an input or output on a within the scope of the restriction νa is not allowed to be taken outside the scope, and dually, an input or an output on a not within the scope of a restriction νa is not allowed to be taken into its scope. In open terms (members of the homset $C(1, 1)$), holes are not allowed to cross any scoping boundaries.

In order to check whether there exists a structural isomorphism between two arrows s and t it is enough to erase all occurrences of ν, check for the existence of a permutation, reintroduce the instances of ν and check whether the permutation respects their scope. Two 2-cells are equal if and only if their domains and codomains coincide, and their underlying permutations are equal. Moreover, we postulate that an automorphism is the identity 2-cell if and only if its underlying permutation is the identity permutation.

[2] We do not quotient the terms with respect to the commutativity of '$\mathbf{|}$' because it is important not to lose the concrete position of a redex within a term when considering interaction with arbitrary contexts—in contrast, the associativity of '$\mathbf{|}$' plays no role and can be quotiented out.

Thus, there are six automorphisms on $a \mid a \mid a$ but only two on $a \mid \nu a.(a \mid a)$. However, there is an invertible 2-cell $\nu a.(b \mid a) \to b \mid \nu a.a$, induced by the identity permutation, capturing the usual structural congruence rule. Similarly, there are invertible 2-cells $\nu a.b \to b$ and $\nu a.a \mid b \to b \mid \nu a.a$, but no 2-cell $\nu a.a \mid \nu a.a \to \nu a.(a \mid a)$.

Vertical composition of 2-cells in the 2-category is the obvious composition of permutations and horizontal composition of 2-cells is defined as expected.

Reactive system. It is a simple exercise to show that all of the data defines a G-category C. We construct a reactive system $\mathbb{C}_{cal}$ by adding rules $\{ \langle a \mid \overline{a}, \epsilon \rangle \mid a \in A \}$ and taking all contexts to be the set of evaluation contexts. The reader will notice that the resulting reduction relation (obtained by instantiating the rules with all contexts) is as expected.

In order to keep the example as simple as possible, we add neither extra axioms or structural rules which guarantee that the null process ϵ is the identity for parallel composition nor do we require any notions of α-equivalence; we note, however, that any derived operational equivalence we shall consider relates terms which would be equated via such axioms. For instance, any arrow P is related with $\epsilon \mid P$ and any two α-equivalent (closed) terms are related.

The 2-category of computations. We shall now show that a reactive system can be used to generate 2-categories in two relevant ways. The first of the two constructions is the classic one, but it does not have an immediate computational intuition associated with the 2-dimensional structure. In the following, all definitions are parametric w.r.t. a reactive systems $\mathbb{C}$, with components $\langle C, \iota, \mathcal{E}, R \rangle$.

Definition 8 (2-category of interactions). *Let C_i denote the 2-category freely generated from the G-computad $\langle C, R \rangle$.*

Indeed, the 2-cells in C_i are generated freely from the original G-category C and the reaction rules R. Thus, in general, a 2-cell of C_i does not denote a meaningful computation in $\mathbb{C}$ as it allows reduction even in non-evaluation contexts.

Definition 9 (2-category of computations). *Let C_c denote the smallest sub-2-category of C_i which includes the reaction rules R and the cells of the G-category $\mathcal{E}$.*

The 2-cells in C_c are generated by extending the original structural isomorphisms in C with the 2-cells corresponding to computations. It is easy to show that there is a close relationship with Leifer and Milner's reaction relation because we use the 2-category $\mathcal{E}$ of *evaluation* contexts in the construction of C_c.

4 From 2-Categories to Double Categories

In Definition 8 we defined the 2-category of interactions. In this section we shall associate to such a 2-category C a double category $\widehat{C}$ that simulates also the potential reductions of partial redexes in C. We start by recalling a construction which lifts the quartet category approach in order to obtain a double category from a 2-category.

$$\begin{tikzcd} & b \arrow[dr, bend left, "f_2"] \arrow[dd, phantom, "\Downarrow"] & \\ a \arrow[ur, bend left, "f_1"] \arrow[dr, bend right, "g_1"'] & & d \\ & c \arrow[ur, bend right, "g_2"'] & \end{tikzcd} \qquad \begin{tikzcd} a \arrow[r, "f_1"] \arrow[d, "g_1"'] \arrow[dr, phantom, "\Swarrow"] & b \arrow[d, "f_2"] \\ c \arrow[r, "g_2"'] & d \end{tikzcd}$$

Fig. 4. A 2-cell and a tile associated to it.

$$\begin{tikzcd} a \arrow[r, "f_1"] \arrow[d, "g_1"'] \arrow[dr, phantom, "\Swarrow"] & b \arrow[r, "f_3"] \arrow[d, "h"] \arrow[dr, phantom, "\Swarrow"] & m \arrow[d, "f_3"] \\ c \arrow[r, "g_2"'] & d \arrow[r, "g_3"'] & n \end{tikzcd} \qquad \begin{tikzcd} & & m \arrow[dr, bend left, "f_3"] \arrow[dd, phantom, "\Downarrow"] & \\ & b \arrow[ur, bend left, "f_2"] \arrow[dr, "h"] \arrow[dd, phantom, "\Downarrow"] & & n \\ a \arrow[ur, bend left, "f_1"] \arrow[dr, bend right, "g_1"'] & & d \arrow[ur, bend right, "g_3"'] & \\ & c \arrow[ur, bend right, "g_2"'] & & \end{tikzcd}$$

Fig. 5. Horizontal composition, and the corresponding 2-cell.

Definition 10 (Quartet double category). *Let C be a 2-category. The quartet double category $\Box C$ is obtained decomposing each cell as in Fig. 4, and defining horizontal and vertical composition as sketched in Fig. 5.*

A check is enough to guarantee that the resulting structure is indeed a double category, and both vertical and horizontal 1-categories coincide with the category underlying C (even if the exchange law becomes more difficult to prove).

As for Example 2, also the previous construction is folklore, the standard reference being probably [15]. It appears implicitly in recent works on tile bisimilarity [4]. ¿From our perspective, it suggests an automatic generation of a labeled relation (abstracting a double category), starting from an unlabelled one (abstracting a 2-category).

From computads to double categories. The mechanism we propose for synthesising labeled transition systems is an instantiation of the general construction of the quartet category: It takes into account the cells of the original G-computad, closing them with just enough information for obtaining the right closure of the resulting double category.

We shall use the notion of *groupoidal idempushouts* [18] (GIPOs), an extension to G-categories of Leifer and Milner's [11] notion of *idempushout* (IPO), in the central construction of Definition 11. Here we shall briefly recall a definition of (G)IPOs, directing to the Appendix for further results and their use in the theory of reactive systems.

Intuitively, a (G)IPO refers to a commutative (up to an isomorphic 2-cell α) square as illustrated in Fig. 6, in which the arrows $g_1 : b \to d$ and $g_2 : c \to d$ are minimal, in the sense that there is no non-trivial arrow $h : e \to d$ and arrows $h_1 : b \to e$, $h_2 : c \to e$ such that $f_1; h_1$ is (up to an isomorphic 2-cell) $f_2; h_2$, $h_1; h$ is (up to an isomorphic 2-cell) g_1 and $h_2; h$ is (up to an isomorphic 2-cell) g_2. When working with G-categories, these isomorphisms are required to paste together to obtain the original isomorphism α.

Given arbitrary f_1 and f_2, it is usual for categories of contexts to have more than one such closure—i.e., there is more than one (G)IPO that has f_1 and f_2 as its lower components. It turns out that to obtain a (G)IPO one constructs a (bi)pushout in a (pseudo) slice category. Such pushouts have been dubbed (groupoidal) relative pushouts, or (G)RPOs.

$$\begin{tikzcd} a \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "\alpha\,\swarrow"] & b \arrow[d, "g_1"] \\ c \arrow[r, "g_2"'] & d \end{tikzcd}$$

Fig. 6. A GIPO

Example 3. Consider the reactive system $\mathbb{C}_{cal}$ previously defined. The underlying category of terms has GRPOs. Diagram (i), below, is a simple example of a GIPO, while diagram (ii), with $\sigma : \nu a.(a \mid \overline{a}) \mid b \to b \mid \nu a.(a \mid \overline{a})$ the unique 2-cell between these two terms is not, since $- \mid b$ is unnecessary and may be factored out. Diagrams (iii) (where $\tau : a \mid \overline{a} \mid b \mid \overline{a} \to a \mid \overline{a} \mid b \mid \overline{a}$ is the permutation which swaps the two copies of $\overline{a}$) and (iv) are both GIPOs, which illustrates our previous remark that two arrows may have several different minimal closures. Diagram (v) is also an example of a GIPO, which is less interesting since the terms $a \mid \overline{a}$ and b are disjoint.

$$\begin{tikzcd} 0 \arrow[r, "\nu a.(a|\overline{a})"] \arrow[d, "a|\overline{a}"'] \arrow[dr, phantom, "\mathrm{id}\,\swarrow"] & 1 \arrow[d, "-"] \\ 1 \arrow[r, "\nu a.-"'] & 1 \end{tikzcd} \quad \begin{tikzcd} 0 \arrow[r, "\nu a.(a|\overline{a})"] \arrow[d, "a|\overline{a}"'] \arrow[dr, phantom, "\sigma\,\swarrow"] & 1 \arrow[d, "-|b"] \\ 1 \arrow[r, "b|\nu a.-"'] & 1 \end{tikzcd} \quad \begin{tikzcd} 0 \arrow[r, "a|\overline{a}|b"] \arrow[d, "a|\overline{a}"'] \arrow[dr, phantom, "\tau\,\swarrow"] & 1 \arrow[d, "-|\overline{a}"] \\ 1 \arrow[r, "-|b|\overline{a}"'] & 1 \end{tikzcd} \quad \begin{tikzcd} 0 \arrow[r, "a|\overline{a}|b"] \arrow[d, "a|\overline{a}"'] \arrow[dr, phantom, "\mathrm{id}\,\swarrow"] & 1 \arrow[d, "-"] \\ 1 \arrow[r, "-|b"'] & 1 \end{tikzcd} \quad \begin{tikzcd} 0 \arrow[r, "b"] \arrow[d, "a|\overline{a}"'] \arrow[dr, phantom, "\mathrm{id}\,\swarrow"] & 1 \arrow[d, "-|a|\overline{a}"] \\ 1 \arrow[r, "b|-"'] & 1 \end{tikzcd}$$

(i) (ii) (iii) (iv) (v)

Definition 11 (Observational double category). *Let $\mathbb{C} = \langle C, \iota, \mathcal{E}, R\rangle$ be a reactive system. The* observational double category *of $\mathbb{C}$, denoted $\mathcal{O}(\mathbb{C})$, is the smallest sub-double category of the quartet double category $\Box\, C_i$ which includes the double cells*

$$\begin{tikzcd} \iota \arrow[r] \arrow[d] \arrow[dr, phantom, "\rho\,\swarrow"] & \iota \arrow[d, "l"] \\ \iota \arrow[r, "r"'] & a \end{tikzcd} \qquad \begin{tikzcd} a \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "\alpha\,\swarrow"] & b \arrow[d, "g_1"] \\ c \arrow[r, "g_2"'] & d \end{tikzcd}$$

(i) (ii)

where the tiles of type (i) correspond to the rules of R, and the tiles of type (ii) correspond to GIPOs in C, with $g_2 \in \mathcal{E}$.

Remark 1. Notice that while the observational double category is a sub-double category of the quartet double category $\Box\, C_i$, the resulting cells are filled in with 2-cells of C_c, as a consequence of requiring g_2 to be an evaluation context. The advantage of working within $\Box C_i$ is that our congruence results (Corollaries 1 and 2) hold w.r.t. all contexts, not just the evaluation contexts.

Thanks to the properties of GIPOs, it is easy to check that the resulting 1-categories coincide with the category underlying C. Later we will show that the proof of decomposition property can be carried out rather easily for the observational double category because of the above facts. Before proving that the decomposition property holds, though, we introduce the notion of depth of a double-cell.

Definition 12 (Depth of a cell). *A cell in* $O(\mathbb{C})$ has depth *n if it contains n occurrences of* ρ *tiles, defined according to Definition 11 (i.e., the cells* ρ *modelling the rules).*

The definition is meaningful, since the closure of the quartet construction allows for no equivalence between cells containing a different number of such basic cells (while this is not the case for those associated with GIPOs).

Example 4. As an example of a cell of depth 2 in the observational double category which results from the reactive system $\mathbb{C}_{cal}$ previously defined, consider the cell illustrated in the diagram below left, which factorises into the rules ρ, ρ' and GIPOs obtained by taking the unique choices for α and α'.

$$
\begin{tikzcd}
\iota \arrow[r, "a|\overline{b}"] \arrow[d] \arrow[dr, phantom, "\xi"] & 1 \arrow[d, "-|b|\overline{a}"] \\
\iota \arrow[r, "\epsilon|\epsilon"'] & 1
\end{tikzcd}
\qquad
\begin{tikzcd}
\iota \arrow[rr] \arrow[d] \arrow[drr, phantom, "\rho"] & & \iota \arrow[r, "a|\overline{b}"] \arrow[d, "b|\overline{b}"'] \arrow[dr, phantom, "\alpha"] & 1 \arrow[d, "-|b"] \\
\iota \arrow[r] \arrow[d] \arrow[dr, phantom, "\rho'"] & \iota \arrow[r, "\epsilon"] \arrow[d, "a|\overline{a}"] \arrow[drr, phantom, "\alpha'"] & 1 \arrow[r, "-|a"] & 1 \arrow[d, "-|\overline{a}"] \\
\iota \arrow[r, "\epsilon"'] & 1 \arrow[rr, "\epsilon|-"'] & & 1
\end{tikzcd}
$$

First, we offer an analysis of the labels of the observational double category. The following lemma is similar in nature to Melliès' Verticalization Theorem [12, Theorem 2] and states that any cell can be decomposed into 'elementary' cells – that is, cells which result from the composition of a single reduction (diagram (*i*) of Definition 11) with a minimal context (diagram (*ii*) of Definition 11).

Lemma 1 (Characterisation). *Let* $f \underset{u}{\longrightarrow} g$ *be a cell of depth n in* $O(\mathbb{C})$*. Then*

- *either* $n \geq 1$ *and there exists cells* $f_{i-1} \underset{u_i}{\longrightarrow} f_i$ *of depth 1 for* $i = 1 \ldots n$ *with* $f_0 = f$ *and* $f_n = g$*, such that* $u = u_1; \ldots; u_n$*;*
- *or* $n = 0$ *and u is an equivalence and* $f, g : \iota \to a$ *are related by an invertible cell in* C_c*.*

The special case for $n = 0$ is a basic consequence of the fact that the square below is always a GIPO, for any invertible cell α relating f and g in C_c.

$$
\begin{tikzcd}
\iota \arrow[r, "f"] \arrow[d] \arrow[dr, phantom, "\alpha"] & a \arrow[d, "id_a"] \\
\iota \arrow[r, "g"'] & a
\end{tikzcd}
$$

Next we can prove the key result.

Lemma 2 (Ground decomposition). $O(\mathbb{C})$ *satisfies ground decomposition.*

Proof. By induction on the depth of a cell $\tau : s \underset{h}{\longrightarrow} t$. If τ has depth 0 then decomposition holds by the decomposition properties of GIPOs (see Lemma 5 in the Appendix). Suppose τ has depth $n > 0$, then by Lemma 1 τ decomposes as shown in Fig. 7(i), where (*a*) α and β are GIPOs; (*b*) ρ models a rewrite rule; and (*c*) τ' has depth $n - 1$.

$$
\begin{tikzcd}
\iota \arrow[rr, "s_1"] \arrow[d] & & a_1 \arrow[r, "s_2"] \arrow[d, phantom, "\alpha"] & a \arrow[d, "h_1"] \\
\iota \arrow[r] \arrow[d] \arrow[dr, phantom, "\rho"] & \iota \arrow[rr, "f"] \arrow[d, "l"] & \arrow[d, phantom, "\beta"] & b \arrow[d, "h_2"] \\
\iota \arrow[r, "r"'] \arrow[d] & e \arrow[rr, "g"] \arrow[d, phantom, "\tau'"] & & c \arrow[d, "k"] \\
\iota \arrow[rrr, "t"'] & & & d
\end{tikzcd}
\qquad
\begin{tikzcd}
\iota \arrow[r, "s_1"] \arrow[d] \arrow[dr, phantom, "\alpha_1"] & a_1 \arrow[r, "s_2"] \arrow[d, dashed, "h'_1"] \arrow[dr, phantom, "\alpha_2"] & a \arrow[d, "h_1"] \\
\iota \arrow[r, "f_1"] \arrow[d, "l"'] \arrow[dr, phantom, "\beta_1"] & b_1 \arrow[r, "f_2"] \arrow[d, dashed, "h'_2"] \arrow[dr, phantom, "\beta_2"] & b \arrow[d, "h_2"] \\
e \arrow[r, "g_1"'] & c_1 \arrow[r, "g_2"'] & c
\end{tikzcd}
$$

(i) (ii)

Fig. 7. Ground decomposition, diagrammatically

Using the fact that GIPOs decompose (see Lemma 5 in the Appendix), we obtain α_1, α_2 such that $\alpha_1 * \alpha_2 = \alpha$ and β_1, β_2 such that $\beta_1 * \beta_2 = \beta$ (see Fig. 7(ii)). The remainder of the decomposition follows via the inductive hypothesis on τ' (along the decomposition of its vertical source $r; g$ in $r; g_1$ and g_2). $\square$

Depth-preserving bisimulation. We exploit the definition of depth in order to offer a refined notion of tile bisimulation.

Definition 13 (Depth-preserving tile bisimulation). *A (ground) tile bisimulation B on $O(\mathbb{C})$* is depth-preserving *if whenever $s\ B\ t$ for $s, t \in C$, then for any cell $s \underset{v}{\longrightarrow} s'$* of depth *$n$ there exists $t' \in C$ and a cell $t \underset{v}{\longrightarrow} t'$* of the same depth *such that $s'\ B\ t'$.*

We shall denote the largest depth-preserving tile bisimulation by $\sim$ and refer to it as *depth-preserving bisimilarity*. It yields Leifer and Milner's semantics for a given reactive system; we include a definition of the latter in the Appendix (Definition 14).

Lemma 3. *Depth preserving tile bisimilarity on $O(\mathbb{C})$ defines the same relation as strong bisimilarity on* $\mathrm{LTS}(\mathbb{C})$ *(as defined in Definition 14).*

Indeed, as a direct consequence of Lemma 2 we have the following corollary.

Corollary 1. *Depth preserving tile bisimilarity on $O(\mathbb{C})$ is a congruence.*

Tile bisimilarity as a weak bisimulation. We shall now look at the results of considering ordinary tile bisimilarity and, in particular, we shall argue that it amounts to a notion of weak bisimulation. This follows straightforwardly from Lemma 1.

Thus, in the bisimulation game, a minimal context which sets off a chain of reactions on f may be matched by the minimal context for another chain of reactions as long as the results are a bisimilar pair of terms. The fact that internal reaction (i.e. only the identity context is provided) can be matched either by internal reaction or no reaction is reminiscent of Milner's original formulation of weak bisimilarity for CCS in which a τ action can be matched by *zero or more* τ's.

Jensen has carried out a preliminary study [9] of defining the notion of weak bisimilarity for reactive systems, and specifically, for bigraphs [10]. We plan to study the relationship between the two bisimilarities as future work.

By definition of depth preserving tile bisimilarity we have the result below.

Lemma 4. *Depth preserving tile bisimilarity $\sim$ on an observational double category implies tile bisimilarity $\approx$ over that double category.*

The case study which follows shows that, in general, $\sim$ is strictly finer than $\approx$ (see hence Example 5). As a direct consequence of Lemma 2 we have the corollary below.

Corollary 2. *Tile bisimilarity $\approx$ on an observational double category is a congruence.*

We conclude by illustrating how the constructions we have seen so far can be applied to the simple process algebra previously introduced.

Example 5. Let $\mathbb{C}_{cal}$ be the reactive system defined in our running example, and let $a \in A$ be a name. Then $\nu a.(a \mid \overline{a}) \nsim 0$ while $\nu a.(a \mid \overline{a}) \approx 0$. For the first part, note that $\nu a.(a \mid \overline{a}) \xrightarrow[\mathrm{id}]{} \nu a.0$ via a cell of depth 1, which cannot be matched by 0. For the second part, observe instead that $\nu a.(a \mid \overline{a}) \xrightarrow[\mathrm{id}]{} \nu a.0$ can be matched by the depth 0 cell $0 \xrightarrow[\mathrm{id}]{} 0$.

5 Conclusions

In this paper we presented a novel approach to the synthesis of a labelled transition system out of reactive system. Our proposal builds on the results by Leifer and Milner, later refined by Sassone and the fourth author, since in order to obtain the contexts necessary for the observation we rely on (groupoidal) relative pushouts. However, we dispense with any set-theoretical presentation. We show instead how the mechanism of synthesising can be obtained as an instance of the classical construction of the quartet category, relating 2-categories and double categories, considered as abstract presentations for reactive and labelled transition systems, respectively.

Our work was also inspired by a series of papers on tile logic, the proof-theoretic counterpart of double categories. The associated tile bisimulation often fails to be a congruence, and the research focused on the characterisation of syntactical constraints for proving when the property holds. One approach has been the saturation of the category with additional cells, thus recovering e.g. (ground) dynamic congruence [4]. The methodology has been applied for recovering s-semantics for logic programming [3].

We feel confident that our contribution streamlines former results on tile logic and synthesised labelled transition systems, and highlights what we consider the basic ingredient on both approaches, namely, the *decomposition property*. In fact, relative pushouts decompose, and this is the reason why the bisimulation on the observational double category is a congruence. Note also that both [3, 4] can be considered as instances of the general approach proposed in the paper since all pushouts are also IPOs.

Future directions. We envision two clear roads for further development. First of all, we would like to tackle open tile bisimulation, in order to lift the restriction to ground reactive systems since, after all, usually a presentation is given in terms of under-specified components, which should be also instantiated, besides being contextualised. Groundness is clearly effective for proving that a bisimulation is a congruence, but of course double categories, with their obvious notion of triggering for a cell, seem to offer a mathematically sound environment where to consider the most general case of open systems. After all, the quartet construction has no restriction whatsoever, and in fact it has a much more general, and stimulating, theory underlying it [15].

The second path to follow concerns the chance of synthesising adequate *concurrent semantics*. Usually, the concurrent semantics for a reactive system is obtained by considering some notion of *permutation equivalence* on reductions; while on labelled transition systems it is usually recovered by capturing some notion of independence on the labels. Thus, the quartet construction appears to be a general mechanism that is well-suited, especially if categories with structure (i.e. either monoidal or cartesian categories) are considered, and the structure on the arrows is lifted to the observations. After all, tile logic has been successfully applied to term and graph rewriting, and the decomposition property has been established in many different settings [1, 6, 12].

References

1. R. Bruni, D. de Frutos-Escrig, N. Martí-Oliet, and U. Montanari. Bisimilarity congruences for open terms and term graphs via tile logic. In *Proc. of CONCUR 2000*, vol. 1877 of *Lect. Notes in Comp. Sci.*, pp. 259–274. Springer, 2000.
2. R. Bruni, J. Meseguer, and U. Montanari. Symmetric and cartesian double categories as a semantic framework for tile logic. *Mathematical Structures in Computer Science*, 12:53–90, 2002.
3. R Bruni, U. Montanari, and F. Rossi. An interactive semantics of logic programming. *Theory and Practice of Logic Programming*, 1:647–690, 2001.
4. R. Bruni, U. Montanari, and V. Sassone. Observational congruences for dynamically reconfigurable tile systems *Theor. Comp. Sci.*, 335(2-3):331-372, 2005.
5. A. Corradini and F. Gadducci. Rewriting on cyclic structures: Equivalence between the operational and the categorical description. *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 33:467–493, 1999.
6. G. Ferrari and U. Montanari. Tile formats for located and mobile systems. *Inform. and Comput.*, 156:173–235, 2000.
7. F. Gadducci, R. Heckel, and M. Llabrés. A bi-categorical axiomatisation of concurrent graph rewriting. In *Proc. of CTCS'99*, vol. 29 of *Electr. Notes in Theor. Comp. Sci.*, Elsevier, 1999.
8. F. Gadducci and U. Montanari. The tile model. In *Proof, Language and Interaction: Essays in Honour of Robin Milner*, pp. 133–166. MIT Press, 2000.
9. O. H. Jensen. Bigraphs and weak bisimilarity. Talk at Dagstuhl Seminar 04241, June 2004.
10. O. H. Jensen and R. Milner. Bigraphs and mobile processes. Technical Report 570, Computer Laboratory, University of Cambridge, 2003.
11. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *Proc. of CONCUR 2000*, vol. 1877 of *Lect. Notes in Comp. Sci.*, pp. 243–258. Springer, 2000.
12. P. A. Melliès. Double categories: A modular model of multiplicative linear logic. *Mathematical Structures in Computer Science*, 12:449–479, 2002.
13. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theor. Comp. Sci.*, 96:73–155, 1992.
14. R. Milner. The polyadic π-calculus: A tutorial. In *Logic and Algebra of Specification*, vol. 94 of *Nato ASI Series F*, pp. 203–246. Springer, 1993.
15. P.H. Palmquist. The double category of adjoint squares. In *Midwest Category Seminar*, vol. 195 of *Lectures Notes in Mathematics*, pp. 123–153. Springer, 1971.
16. A.J. Power. An abstract formulation for rewrite systems. In *Category Theory and Computer Science*, vol. 389 of *Lect. Notes in Comp. Sci.*, pp. 300–312. Springer, 1989.
17. D.E. Rydehard and E.G. Stell. Foundations of equational deductions: A categorical treatment of equational proofs and unification algorithms. In *Category Theory and Computer Science*, vol. 283 of *Lect. Notes in Comp. Sci.*, pp. 114–139. Springer, 1987.

18. V. Sassone and P. Sobociński. Deriving bisimulation congruences using 2-categories. *Nordic Journal of Computing*, 10:163–183, 2003.
19. P. Sewell. From rewrite rules to bisimulation congruences. *Theor. Comp. Sci.*, 274:183–230, 2004.
20. R.H. Street. Categorical structures. *Handbook of Algebra*, vol. 1, pp. 529–577. North-Holland, 1996.

Appendix: (G)IPOs

Using the universal properties of (bi)colimits, one can prove that (G)IPOs satisfy several basic properties reminiscent of ordinary pushouts.

Lemma 5 (Composition and decomposition of (G)IPOs). *Let C be a (G-)category which has (G)RPOs. Then*

$$\begin{tikzcd} a \arrow[r, "f_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "\swArrow\alpha"] & b \arrow[r, "f_1'"] \arrow[d, "g_1"] \arrow[dr, phantom, "\swArrow\sigma"] & b' \arrow[d, "g_1'"] \\ c \arrow[r, "g_2"'] & d \arrow[r, "g_2'"'] & d' \end{tikzcd} \qquad \begin{tikzcd} a \arrow[r, "f_1;f_1'"] \arrow[d, "f_2"'] \arrow[dr, phantom, "\swArrow f_1;\sigma \bullet \alpha;g_2'"] & Z \arrow[d, "g_1'"] \\ c \arrow[r, "g_2;g_2'"'] & d' \end{tikzcd}$$

(i) $\qquad$ (ii)

1. *if both squares α and β in diagram (i) are (G)IPOs then the exterior (see diagram (ii)) is also a (G)IPO;*
2. *if the left square α and the exterior (see diagram (ii)) of diagram (i) are (G)IPOs then so is the right square.*

The basic idea, originally due to Sewell [19], is that the labels are the smallest contexts which allow a reaction to occur.

Definition 14 (LTS). *Let $\mathbb{C}$ be a reactive system. The associated labelled transition systems* LTS($\mathbb{C}$) *is given by*

1. *the states of* LTS($\mathbb{C}$) *are arrows $s\colon \iota \to a$ in C*
2. *there is a transition $s \xrightarrow{f} t'$ iff there exists $\langle l, r\rangle \in R$, $t \in \mathcal{E}$ and a 2-cell $\alpha : s; f \Rightarrow l; t$ such that the square below is a (G)IPO and $t' = r; t$.*

$$\begin{tikzcd} \iota \arrow[r, "s"] \arrow[d, "l"'] \arrow[dr, phantom, "\swArrow\alpha"] & a \arrow[d, "f"] \\ b \arrow[r, "t"'] & c \end{tikzcd}$$

In the case of G-categories, one normally quotients the states and the transitions of the LTS with respect to isomorphism—in other words, the 2-dimensional structure is no longer necessary and may be discarded.

One of the main results that holds for such an LTS is that when the underlying (G-)category has enough (G)RPOs (one only has to require so-called *redex*-GRPOs to exist), then bisimilarity is a congruence. This was originally shown by Leifer and Milner [11] and extended to the more general setting by Sassone and the fourth author [18].