Deriving Bisimulation Congruences: 2-Categories Vs Precategories *

Vladimiro Sassone[1] and Paweł Sobociński[2]

[1] University of Sussex
[2] University of Aarhus

Abstract. G-relative pushouts (GRPOs) have recently been proposed by the authors as a new foundation for Leifer and Milner's approach to deriving labelled bisimulation congruences from reduction systems. This paper develops the theory of GRPOs further, arguing that they provide a simple and powerful basis towards a comprehensive solution. As an example, we construct GRPOs in a category of 'bunches and wirings.' We then examine the approach based on Milner's precategories and Leifer's functorial reactive systems, and show that it can be recast in a much simpler way into the 2-categorical theory of GRPOs.

Introduction

It is increasingly common for foundational calculi to be presented as *reduction systems*. Starting from their common ancestor, the λ calculus, most recent calculi consist of a reduction system together with a contextual equivalence (built out of basic observations, viz. barbs). The strength of such an approach resides in its intuitiveness. In particular, we need not invent labels to describe the interactions between systems and their possible environments, a procedure that has a degree of arbitrariness (cf. early and late semantics of the π calculus) and may prove quite complex (cf. [5, 4, 3, 1]).

By contrast, reduction semantics suffer at times by their lack of compositionality, and have complex semantic theories because of their contextual equivalences. Labelled bisimulation congruences based on *labelled transition systems* (LTS) may in such cases provide fruitful proof techniques; in particular, bisimulations provide the power and manageability of coinduction, while the closure properties of congruences provide for compositional reasoning.

To associate an LTS with a reduction system involves synthesising a compositional system of labels, so that silent moves (or τ-actions) reflect the original reductions, labels describe potential external interactions, and all together they yield a LTS bisimulation which is a congruence included in the original contextual reduction equivalence. Proving bisimulation is then enough to prove reduction equivalence.

Sewell [19] and Leifer and Milner [13, 11] set out to develop a theory to perform such derivations using general criteria; a meta-theory of *deriving bisimulation congruences*. The basic idea behind their construction is to use contexts as labels. To exemplify the idea, in a CCS-like calculus one would for instance derive a transition

* Research supported by '**DisCo**: Semantic Foundations of Distributed Computation', EU IHP 'Marie Curie' contract HPMT-CT-2001-00290, and **BRICS**, Basic Research in Computer Science, funded by the Danish National Research Foundation.

A.D. Gordon (Ed.): FOSSACS 2003, LNCS 2620, pp. 409–424, 2003.

$$a.P \xrightarrow{-\mid \bar{a}.Q} P \mid Q$$

because term $a.P$ in context $- \mid \bar{a}.Q$ reacts to become $P \mid Q$; in other words, the context is a trigger for the reduction.

The first hot spot of the theory is the selection of the right triggers to use as labels. The intuition is to take only the "*smallest*" contexts which allow a given reaction to occur. As well as reducing the size of the LTS, this often makes the resulting bisimulation equivalence finer. Sewell's method is based on dissection lemmas which provide a deep analysis of a term's structure. A generalised, more scalable approach was later developed in [13], where the notion of "smallest" is formalised in categorical terms as a *relative-pushout* (RPOs). Both theories, however, do not seem to scale up to calculi with non trivial *structural congruences*. Already in the case of the monoidal rules that govern parallel composition things become rather involved.

The fundamental difficulty brought about by a structural congruence $\equiv$ is that working up to $\equiv$ gives up too much information about terms for the RPO approach to work as expected. RPOs do not usually exist in such cases, because the fundamental indication of exactly which occurrences of a term constructor belong to the redex becomes blurred. A very simple, yet significant example of this is the category **Bun** of bunch contexts [13], and the same problems arise in structures such as action graphs [14] and bigraphs [15].

In [17] we therefore proposed a framework in which term structure is not explicitly quotiented, but the commutation of diagrams (i.e. equality of terms) is taken up to $\equiv$. Precisely, to give a commuting diagram $rp \equiv sq$ one exhibits a proof α of structural congruence, which we represent as a 2-cell (constructed from the rules generating $\equiv$ and closed under all contexts).

$$\begin{array}{ccc} k & \xrightarrow{p} & l \\ {\scriptstyle q}\downarrow & \alpha & \downarrow{\scriptstyle r} \\ m & \xrightarrow[s]{} & n \end{array}$$

Since such proofs are naturally isomorphisms, we were led to consider G-categories, i.e., 2-categories where all 2-cells are iso, and initiated the study of G-relative pushouts (GRPOs), as a suitable generalisation of RPOs from categories to G-categories.

The purpose of this paper is to continue the development of the theory of GRPOs. We aim to show that, while replacing RPOs at little further complication (cf. §1 and §2), GRPOs significantly advance the field by providing a convenient solution to simple, yet important problems (cf. §3 and §4). The theory of GRPOs promises indeed to be a natural foundation for a meta-theory of 'deriving bisimulation congruences.'

This paper presents two main technical results in support of our claims. Firstly, we prove that the case of the already mentioned category **Bun** of bunch contexts, problematic for RPOs, can be treated in a natural way using GRPOs. Secondly, we show that the notions of precategory and functorial reactive system can be dispensed with in favour of a simpler GRPO-based approach.

The notion of *precategory* is proposed in [11, 12] to handle the examples of Leifer in [11], Milner in [15] and, most recently, of Jensen and Milner in [7]. It consists of a

category appropriately decorated by so-called "*support sets*" which identifies syntactic elements so as to keep track of them under arrow composition. Alas, such supported structures are no longer categories – arrow composition is partial – which makes the theory laborious, and bring us away from the well-known world of categories and their theory. The intensional information recorded in precategories, however, allows one to generate a category "above" where RPOs exist, as opposed to the category of interest "below", say $\mathbb{C}$, where they do not. The category "above" is related to $\mathbb{C}$ via a well-behaved functor, used to map RPOs diagrams from the category "above" to $\mathbb{C}$, where constructing them would be impossible. These structures take the name of *functorial reactive systems*, and give rise to a theory to generate a labelled bisimulation congruences developed in [11].

The paper presents a technique for mapping precategories to G-categories so that the LTS generated using GRPOs is the same as the LTS generated using the above mentioned approach. The translation derives from the precategory's support information a notion of homomorphism, specific to the particular structure in hand, which constitutes the 2-cells of the derived G-category. We claim that this yields an approach mathematically more elegant and considerably simpler than precategories; besides generalising RPOs directly, GRPOs seem to also remove the need for further notions.

Structure of the paper. In §1 we review definitions and results presented in [17]; §2 shows that, analogously to the 1-dimensional case, trace and failures equivalence are congruences provided that enough GRPOs exist. In §3, we show that the category of bunch contexts is naturally a 2-category where GRPOs exist; §4 shows how precategories are subsumed by our notion of GRPOs. Most proofs in this extended abstract are either omitted or sketched. For these, the interested reader should consult [18].

1 Reactive Systems and GRPOs

Lawvere theories [10] provide a canonical way to recast term algebras as categories. For Σ a signature, the (free) Lawvere theory on Σ, say $\mathbf{C}_\Sigma$, has the natural numbers for objects and a morphism $t\colon m \to n$, for t a n-tuple of m-holed terms. Composition is substitution of terms into holes.

Generalising from term rewriting systems on $\mathbf{C}_\Sigma$, Leifer and Milner formulated a definition of *reactive system* [13], and defined a technique to extract labelled bisimulation congruences from them. In order to accommodate calculi with non trivial structural congruences, as explained in the Introduction, we refine their approach as follows.

Definition 1.1. A G-***category*** is a 2-category where all 2-cells are isomorphisms.

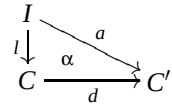
A G-category is a thus a category enriched over G, the category of groupoids.

Definition 1.2. A G-***reactive system*** $\mathbf{C}$ consists of a G-category $\mathbb{C}$; a subcategory $\mathbb{D}$ of *reactive contexts*, required to be closed under 2-cells and composition-reflecting; a distinguished object $I \in \mathbb{C}$; a set of pairs $R \subseteq \bigcup_{C\in\mathbb{C}} \mathbb{C}(I,C) \times \mathbb{C}(I,C)$, called the *reaction rules*.

The reactive contexts are those contexts inside which evaluation may occur. By composition-reflecting we mean that $dd' \in \mathbb{D}$ implies d and $d' \in \mathbb{D}$, while the closure property means that given $d \in \mathbb{D}$ and $\rho\colon d \Rightarrow d'$ in $\mathbb{C}$ implies $d' \in \mathbb{D}$. The reaction relation $\longrightarrow\!\!\triangleright$ is defined by taking

$$a \longrightarrow\!\!\triangleright\, dr \quad \text{if there exists } \langle l, r\rangle \in R\,,\ d \in \mathbb{D} \text{ and } \alpha\colon dl \Rightarrow a \text{ in } \mathbb{C}$$

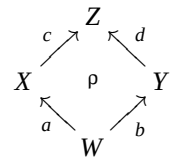
As illustrated by the diagram below, this represents the fact that, up to structural congruence, a is the left-hand side l of a reduction rule in a reaction context d.



The notion of GRPO formalises the idea of a context being the "smallest" that enables a reaction in a G-reactive system, and is a conservative 2-categorical extension of Leifer and Milner RPOs [13] (cf. [17] for a precise comparison).

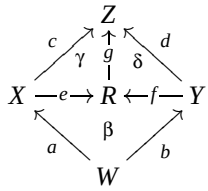
For readers acquainted with 2-dimensional category theory (cf. [9] for a thorough introduction), GRPOs are defined in Definition 1.3. This is followed by an elementary presentation in Proposition 1.4 taken from [17]. We use $\bullet$ for vertical composition.

Definition 1.3 (GRPOs). Let $\rho\colon ca \Rightarrow db\colon W \to Z$ be a 2-cell (see diagram below) in a G-category $\mathbb{C}$. A G-***relative pushout*** (GRPO) for ρ is a bipushout (cf. [8]) of the pair or arrows $(a,\mathbf{1}) : ca \to c$ and $(b,\rho) : ca \to d$ in the pseudo-slice category $\mathbb{C}/Z$.

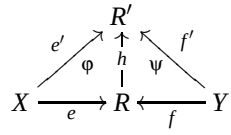


(1)

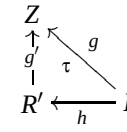
Proposition 1.4. Let $\mathbb{C}$ be a G-category. A candidate GRPO for $\rho\colon ca \Rightarrow db$ as in diagram (1) is a tuple $\langle R, e, f, g, \beta, \gamma, \delta\rangle$ such that $\delta b \bullet g\beta \bullet \gamma a = \rho$ – cf. diagram (*i*).



(*i*)



(*ii*)



(*iii*)

A GRPO for ρ is a candidate which satisfies a universal property. Namely, for any other candidate $\langle R', e', f', g', \beta', \gamma', \delta'\rangle$ there exists a quadruple $\langle h, \varphi, \psi, \tau\rangle$ where $h\colon R \to R'$, $\varphi\colon e' \Rightarrow he$ and $\psi\colon hf \Rightarrow f'$ – cf. diagram (*ii*) – and $\tau\colon g'h \Rightarrow g$ – diagram (*iii*) – which makes the two candidates compatible in the obvious way, i.e.

$$\tau e \bullet g'\varphi \bullet \gamma' = \gamma \qquad \delta' \bullet g'\psi \bullet \tau^{-1} f = \delta \qquad \psi b \bullet h\beta \bullet \varphi a = \beta'.$$

Such a quadruple, which we shall refer to as *mediating morphism*, must be *essentially unique*. Namely, for any other mediating morphism $\langle h', \varphi', \psi', \tau' \rangle$ there must exist a *unique* two cell $\xi\colon h \to h'$ which makes the two mediating morphisms compatible, i.e.

$$\xi e \bullet \varphi = \varphi' \qquad \psi \bullet \xi^{-1} f = \psi' \qquad \tau' \bullet g' \xi = \tau.$$

Observe that whereas RPOs are defined up to isomorphism, GRPOs are defined up to equivalence (since they are bicolimits).

The definition below plays an important role in the following development.

Definition 1.5 (GIPO). Diagram (1) of Definition 1.3 is said to be a G-***idem-pushout*** (GIPO) if $\langle Z, c, d, \mathrm{id}_Z, \rho, \mathbf{1}_c, \mathbf{1}_d \rangle$ is its GRPO.

We recall in §A the essential properties of GRPOs and GIPOs from [17].

Definition 1.6 (LTS). For $\mathbf{C}$ a G-reactive system whose underlying category $\mathbb{C}$ is a G-category, define $\mathsf{GTS}(\mathbf{C})$ as follows:

- the states $\mathsf{GTS}(\mathbf{C})$ are iso-classes of arrows $[a]\colon I \to X$ in $\mathbf{C}$;
- there is a transition $[a] \xrightarrow{[f]} [a']$ if there exists a 2-cell ρ, a rule $\langle l, r \rangle \in R$, and $d \in \mathbb{D}$ with $a' \cong dr$ and such that the diagram below is a GIPO.

$$\begin{array}{ccccc} & & Z & & \\ & f \nearrow & & \nwarrow d & \\ X & & \rho & & Y \\ & a \nwarrow & & \nearrow l & \\ & & I & & \end{array} \tag{2}$$

Henceforward we shall abuse notation and leave out the square brackets when writing transitions; ie. we shall write simply $a \xrightarrow{f} a'$ instead of $[a] \xrightarrow{[f]} [a']$.

Categories can be seen as a discrete G-categories (the only 2-cells are identities). Using this observation, each G-concepts introduced above reduces to the corresponding 1-categorical concept. For instance, a GRPO in a category is simply a RPO.

2 Congruence Results for GRPOs

The fundamental property that endows the LTS derived from a reduction system with a bisimulation which is a congruence is the following notion.

Definition 2.1 (Redex GRPOs). A G-reactive system $\mathbf{C}$ is said to ***have redex GRPOs*** if every square (2) in its underlying G-category $\mathbb{C}$ with l the left-hand side of a reaction rule $\langle l, r \rangle \in R$, and $d \in \mathbb{D}$ has a GRPO.

In particular, the main theorem of [17] is as follows.

Theorem 2.2 (cf. [17]). *Let* $\mathbf{C}$ *be a reactive system whose underlying* G*-category* $\mathbb{C}$ *has redex* G*RPOs. The largest bisimulation* $\sim$ *on* $\mathsf{GTS}(\mathbf{C})$ *is a congruence.*

The next three subsections complement this result by proving the expected corresponding theorems for trace and failure semantics, and by lifting them to the case of weak equivalences. Theorems and proofs in this section follow closely [11], as they are meant to show that GRPOs are as viable a tool as RPOs are.

2.1 Traces Preorder

Trace semantics [16] is a simple notion of equivalence which equates processes if they can engage in the same sequences of actions. Even though it lacks the fine discriminating power of branching time equivalences, viz. bisimulations, it is nevertheless interesting because many safety properties can be expressed as conditions on sets of traces.

We say that a sequence $f_1 \cdots f_n$ of labels of $\mathsf{GTS}(\mathbf{C})$ is a trace of a if

$$a \xrightarrow{f_1} \cdots \xrightarrow{f_n} a_{n+1}$$

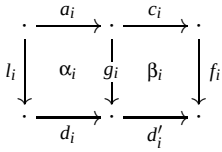
for some $a_1, \ldots, a_n$. The trace preorder $\lesssim_{\mathrm{tr}}$ is then defined as $a \lesssim_{\mathrm{tr}} b$ if all traces of a are also traces of b.

Theorem 2.3 (Trace Congruence). *$\lesssim_{\mathrm{tr}}$ is a congruence.*

Proof. Assume $a \lesssim_{\mathrm{tr}} b$. We prove that $ca \lesssim_{\mathrm{tr}} cb$ for all contexts $c \in \mathbb{C}$. Suppose that

$$ca = \overline{a}_1 \xrightarrow{f_1} \overline{a}_2 \cdots \overline{a}_n \xrightarrow{f_n} \overline{a}_{n+1}.$$

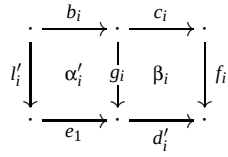
We first prove that there exist a sequence, for $i = 1, \ldots, n$,



where $a_1 = a$, $c_1 = c$, $c_{i+1} = d'_i$, $\overline{a}_i = c_i a_i$, and each square is a GIPO.[1] The ith induction step proceeds as follows. Since $\overline{a}_i \xrightarrow{f_i} \overline{a}_{i+1}$, there exists $\gamma_i \colon f_i c_i a_i \Rightarrow \overline{d}_i l_i$, for some $\langle l_i, r_i \rangle \in R$ and $\overline{d}_i \in \mathbb{D}$, with $\overline{a}_{i+1} = \overline{d}_i r_i$. Since $\mathbf{C}$ has redex GIPOs (cf. Definition 2.1), this can be split in two GIPOs: $\alpha_i \colon g_i a_i \Rightarrow d_i l_i$ and $\beta_i \colon f_i c_i \Rightarrow d'_i g_i$ (cf. diagram above). Take $a_{i+1} = d_i r_i$, and the induction hypothesis is maintained. In particular, we obtained a trace

$$a = a_1 \xrightarrow{g_1} a_2 \cdots a_n \xrightarrow{g_n} a_{n+1}$$

that, in force of the hypothesis $a \lesssim_{\mathrm{tr}} b$ must be matched by a corresponding trace of b. This means that, for $i = 1, .., n$, there exist GIPOs $\alpha'_i \colon g_i b_i \Rightarrow e_i l'_i$, for some $\langle l'_i, r'_i \rangle \in R$ and $e_i \in \mathbb{D}$, once we take b_{i+1} to be $e_i r'_i$. We can then paste each of such GIPOs together with $\beta_i \colon f_i c_i \Rightarrow d'_i g_i$ obtained above and, using Lemma A.3, conclude that there exist GIPOs $f_i c_i b_i \Rightarrow d'_i e_i l'_i$, as in the diagram below.



which means $c_i b_i \xrightarrow{f_i} d'_i e_i r'_i$.

[1] Since the fact is not likely to cause confusion, we make no notational distinction between the arrows of $\mathbf{C}$ (e.g. in GRPOs diagrams) and the states and labels of $\mathsf{GTS}(\mathbf{C})$, where the latter are iso-classes of the former.

As $cb = c_1 b_1$, in order to construct a trace $cb = \bar{b}_1 \xrightarrow{f_1} \cdots \xrightarrow{f_n} \bar{b}_{n+1}$ and complete the proof, we only need to verify that for $i = 1, \ldots, n$, we have that $d'_i e_i r'_i = c_{i+1} b_{i+1}$. This follows at once, as $c_{i+1} = d'_i$ and $b_{i+1} = e_i r'_i$. □

2.2 Failures Preorder

Failure semantics [6] enhances trace semantics with limited branch-inspecting power. More precisely, failure sets allow the testing of when processes renounce the capability of engaging in certain actions.

Formally, for a a state of $\mathsf{GTS}(\mathbf{C})$, a *failure* of a is a pair $(f_1 \cdots f_n, X)$, where $f_1 \cdots f_n$ and X are respectively a sequence and a set of labels, such that:

- $f_1 \cdots f_n$ is a trace of a, $a \xrightarrow{f_1} \cdots \xrightarrow{f_n} a_{n+1}$;
- a_{n+1}, the final state of the trace, is *stable*, i.e. $a_{n+1} \not\longrightarrow\!\triangleright$;
- a_{n+1} *refuses* X, i.e. $a_{n+1} \not\xrightarrow{x}$ for all $x \in X$.

The failure preorder $\lesssim_{\mathrm{f}}$ is defined as $a \lesssim_{\mathrm{f}} b$ if all failures of a are also failures of b. The proof of the following result can be found in [18].

Theorem 2.4 (Failures Congruence). *$\lesssim_{\mathrm{f}}$ is a congruence.*

2.3 Weak Equivalences

Theorems 2.2, 2.3, and 2.4 can be extended to weak equivalences, as outlined below.

For f a label of $\mathsf{GTS}(\mathbf{C})$ define a *weak transition* $a \overset{f}{\Longrightarrow} b$ to be a mixed sequence of transitions and reductions $a \longrightarrow\!\triangleright^* \xrightarrow{f} \longrightarrow\!\triangleright^* b$. Observe that this definition essentially identifies silent transitions in the LTS with reductions. As a consequence, care has to be taken to avoid interference with transitions $\xrightarrow{equi}$ synthesised from GRPOs and labelled by an equivalence. These transitions have essentially the same meaning as silent transitions (i.e. no context involved in the reduction), and must therefore be omitted in weak observations. This lead to consider the following definitions.

Definition 2.5 (Weak Traces and Failures). A sequence $f_1 \cdots f_n$ of *non-equivalence* labels of $\mathsf{GTS}(\mathbf{C})$ is a weak trace of a if

$$a \overset{f_1}{\Longrightarrow} a_1 \cdots a_{n-1} \overset{f_n}{\Longrightarrow} a_n$$

for some $a_1, \ldots, a_n$. The weak trace preorder is then defined accordingly.

A *weak failure* of a is a pair $(f_1 \cdots f_n, X)$, where $f_1 \cdots f_n$ and X are respectively a sequence and a set of *non-equivalence* labels, such that $f_1 \cdots f_n$ is a weak trace of a reaching a final state which is stable and refuses X. The weak trace preorder is defined accordingly.

Definition 2.6 (Weak Bisimulation). A symmetric relation S on $\mathsf{GTS}(\mathbf{C})$ is a weak bisimulation if for all $a \; S \; b$

$$a \xrightarrow{f} a' \quad f \text{ not an equivalence}, \quad \text{implies } b \overset{f}{\Longrightarrow} b' \text{ with } a' \; S \; b'$$
$$a \longrightarrow\!\triangleright\; a' \quad \text{implies } b \longrightarrow\!\triangleright^* b' \text{ with } a' \; S \; b'$$

Using the definitions above Theorems 2.2, 2.3, and 2.4 can be lifted, respectively, to weak traces, failures and bisimulation.

It is worth remarking that the congruence results, however, only hold for contexts $c \in \mathbb{D}$, as it is well known that non reactive contexts (i.e. those c where $ca \longrightarrow\!\!\triangleright\, cb$ does not follow from $a \longrightarrow\!\!\triangleright\, b$, as e.g. the CSS context $c = c_0 + -$) do not preserve weak equivalences. Alternative definitions of weak bisimulations are investigated in [11], and they are applicable *mutatis mutandis* to GRPOs.

3 Bunches and Wires

The category of "bunches and wires" was introduced in [13] as a skeletal algebra of shared wirings, abstracting over the notion of *names* in, e.g., the π calculus. Although elementary, its structure is complex enough to lack RPOs.

A bunch context of type $m_0 \to m_1$ consists of an ordered set of m_1 trees of depth 1 containing exactly m_0 holes. Leaves are labelled from an alphabet K.

Definition 3.1. The category of ***bunch contexts*** $\mathbf{Bun}_0$ has

- objects the finite ordinals, denoted $m_0, m_1, \ldots$
- arrows are bunch contexts $c = (X, \mathrm{char}, \mathrm{root}) : m_0 \to m_1$, where X is a finite carrier, $\mathrm{root} : m_0 + X \to m_1$ is a surjective function linking leaves (X) and holes (m_0) to their roots (m_1), and $\mathrm{char} : X \to K$ is a leaf labelling function.

Composing $c_0 : m_0 \to m_1$ and $c_1 : m_1 \to m_2$ means filling the m_1 holes of c_1 with the m_1 trees of c_0. Formally, $c_1 c_0$ is $(X, \mathrm{root}, \mathrm{char})$ where

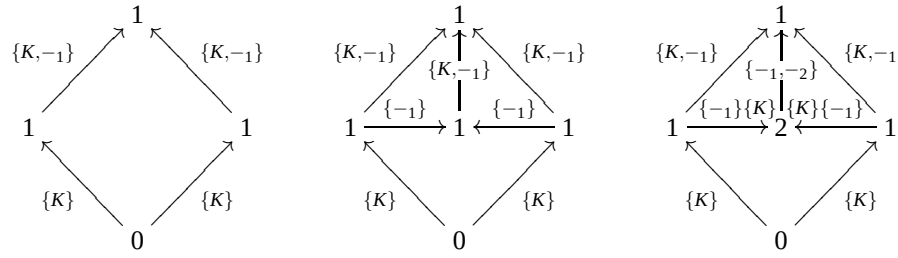
$$X = X_0 + X_1, \qquad \mathrm{root} = \mathrm{root}_1(\mathrm{root}_0 + \mathrm{id}_{X_1}), \qquad \mathrm{char} = [\mathrm{char}_0, \mathrm{char}_1],$$

where $+$ and $[\_,\_]$ are, resp., coproduct and copairing. Identities are $(\emptyset, !, \mathrm{id}) : m_0 \to m_0$.

A ***homomorphism*** of bunch contexts $\rho : c \Rightarrow c' : m_0 \to m_1$ is a function $\rho : X \to X'$ which respects root and char, i.e. $\mathrm{root}' \rho = \mathrm{root}$ and $\mathrm{char}' \rho = \mathrm{char}$. An isomorphism is a bijective homomorphism. Isomorphic bunch contexts are equated, making composition associative and $\mathbf{Bun}_0$ a category.

A bunch context $c : m_0 \to m_1$ can be depicted as a string of m_1 nonempty multisets on $K + m_0$, with the proviso that elements m_0 must appear exactly once in the string. In the examples, we represent elements of m_0 as numbered holes $-_i$.

As we mentioned before, RPOs do not exist in $\mathbf{Bun}_0$. Indeed, consider *(i)* below together with the two candidates *(ii)* and *(iii)*. It is easy to show that these have no common "lower bound" candidate.



The point here is that by taking the arrows of $\mathbf{Bun}_0$ up to isomorphism we lose information about *how* bunch contexts equal each other. Diagram *(i)*, for instance, can be commutative in two different ways: the K in the bottom left part may corresponds either to the one in the bottom right or to the one in the top right, according to whether we read $\{K,-_1\}$ or $\{-_1,K\}$ for the top rightmost arrow. In order to track this information we endow $\mathbf{Bun}_0$ with its natural 2-categorical structure.

Definition 3.2. The 2-category of bunch contexts **Bun** has:

- objects the finite ordinals, denoted $m_0, m_1, \ldots$; we use **Ord** to denote the category of finite ordinals, and $\oplus$ for ordinal addition.
- arrows $c = (x, \mathrm{char}, \mathrm{root})\colon m_0 \to m_1$ consist of a finite ordinal x, a surjective function $\mathrm{root}\colon m_0 \oplus x \to m_1$ and a labelling function $\mathrm{char}\colon x \to K$.
- 2-cells ρ are isomorphisms between bunches' carriers.

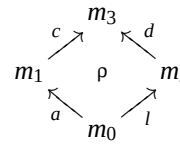
Composition of arrows and 2-cells is defined in the obvious way. Notice that since $\oplus$ is associative, composition in **Bun** is associative. Therefore **Bun** is a G-category.

Replacing the carrier set X with a finite ordinal x allows us to avoid the unnecessary burden of working in a bicategory, which would arise because sum on sets is only associative up to isomorphism. Observe that this simplification is harmless since the set theoretical identity of the elements of the carrier is irrelevant. We remark, however, that GRPOs are naturally a bicategorical notion and would pose no particular challenge in bicategories.

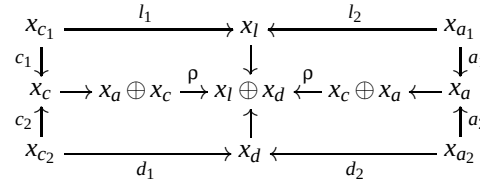
Theorem 3.3. **Bun** has GRPOs.

Proof. Here we give a basic account of the construction of a GRPO, but omit the proof of universality. In the following, we use only the fact that **Bun** is an extensive(cf. [2]) category with pushouts.

Suppose that we have



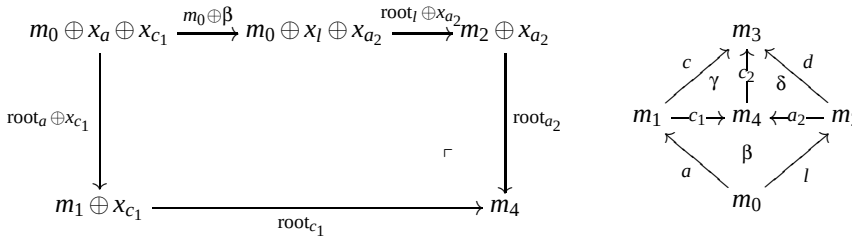
In the following diagram all the rectangles are pullbacks in **Ord** and all the outside arrows are coproduct injections.



Using the morphisms from the diagram above as building blocks, we can construct bijections $\gamma\colon x_c \to x_{c_1} \oplus x_{c_2}$, $\delta\colon x_{a_2} \oplus x_{c_2} \to x_d$ and $\beta\colon x_a \oplus x_{c_1} \to x_l \oplus x_{a_2}$ such that

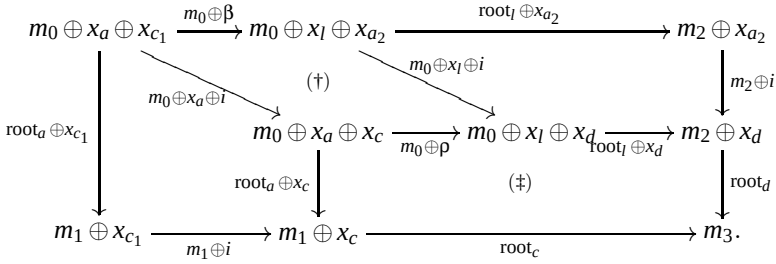
$$x_l \oplus \delta.\beta \oplus x_{c_2}.x_a \oplus \gamma = \rho. \tag{3}$$

Let root_{c_1} and root_{a_2} be the morphisms making *(i)* below



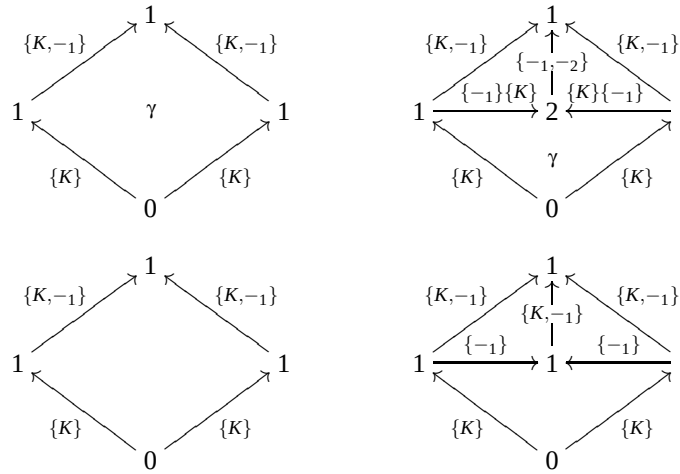
into a pushout diagram. We can define char_{c_1}, char_{a_2} and char_{c_2} in the obvious way.

Now consider the diagram below:



Region $(\dagger)$ can be verified to be commutative using (3) while region $(\ddagger)$ commutes since ρ is a homomorphism. Using the pushout property, we get a unique function $h\colon m_4 \to m_3$. Thus we define $\mathrm{root}_{c_2}\colon m_4 \oplus x_{c_2} \to m_3$ as $[h, \mathrm{root}_c\, i]$. It is easy to verify that this function is surjective. □

Example 3.4. Let $\gamma\colon 2 \to 2$ be the function taking $1 \mapsto 2$ and $2 \mapsto 1$. We give below on the right the GRPOs for the squares on the left.



4 2-Categories Vs Precategories

Other categories which, besides $\mathbf{Bun}_0$, lack RPOs include the closed *shallow action contexts* [11, 12] and *bigraph contexts* [15, 7]. The solution adopted by Leifer [12] and later by Milner [15] is to introduce a notion of a *well-supported precategory*, where the algebraic structures at hand are decorated by finite "support sets." The result is no longer a category – since composition of arrows is defined only if their supports are disjoint – but from any such precategory one can generate two categories which jointly allow the derivation of a bisimulation congruence via a *functorial reactive system*. These categories are the so-called *track* category, where support information is built into the objects, and the *support quotient* category, where arrows are quotiented by the support structure. The track category has enough RPOs and is mapped to the support quotient category via a well-behaved *functor*, so as to transport RPOs adequately.

In this section we present a translation from precategories to G-categories. The main result shows that the LTS derived using precategories and functorial reactive systems is identical to the LTS derived using GRPOs. We begin with a brief recapitulation of the definitions from [12].

Definition 4.1. A ***precategory*** $\mathbb{A}$ consists of the same data as a category. The composition operator $\circ$ is, however, a partial function which satisfies

1. for any arrow $f : A \to B$, $\mathrm{id}_B \circ f$ and $f \circ \mathrm{id}_A$ are defined and $\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A$;
2. for any $f : A \to B$, $g : B \to C$, $h : C \to D$, $(h \circ g) \circ f$ is defined iff $h \circ (g \circ f)$ is defined and then $(h \circ g) \circ f = h \circ (g \circ f)$.

Definition 4.2. Let Set_f be the category of finite sets. A ***well supported precategory*** is a pair $\langle \mathbb{A}, |-| \rangle$, where $\mathbb{A}$ is a precategory and $|-| : \mathrm{Arr}\,\mathbb{A} \to \mathsf{Set}_f$ is the so-called support function, satisfying:

1. $g \circ f$ is defined iff $|g| \cap |f| = \emptyset$, and if $g \circ f$ is defined then $|g \circ f| = |g| \cup |f|$;
2. $|\mathrm{id}_A| = \emptyset$.

For any $f : A \to B$ and any injective function ρ in Set_f the domain of which contains $|f|$ there exists an arrow $\rho \cdot f : A \to B$ called the ***support translation*** of f by ρ. The following axioms are to be satisfied.

1. $\rho \cdot \mathrm{id}_A = \mathrm{id}_A$;
2. $\mathrm{id}_{|f|} \cdot f = f$;
3. $\rho_0|f| = \rho_1|f|$ implies $\rho_0 \cdot f = \rho_1 \cdot f$;
4. $\rho \cdot (g \circ f) = \rho \cdot g \circ \rho \cdot f$;
5. $(\rho_1 \circ \rho_0) \cdot f = \rho_1 \cdot (\rho_0 \cdot f)$;
6. $|\rho \cdot f| = \rho|f|$.

We illustrate these definitions giving a precategorical definition of bunches and wiring (viz. §3).

Example 4.3 (Bunches). The precategory of bunch contexts **A-Bun** has objects and arrows as in $\mathbf{Bun}_0$. However, differently from $\mathbf{Bun}_0$, they are not taken up to isomorphism here. The support of $c = (X, \mathrm{char}, \mathrm{root})$ is X. Composition $c_1 c_0 = (X, \mathrm{char}, \mathrm{root}) : m_0 \to m_2$ of $c_0 : m_0 \to m_1$ and $c_1 : m_1 \to m_2$ is defined if $X_0 \cap X_1 = \emptyset$ and, if so, we have $X = X_0 \cup X_1$. Functions char and root are defined in the obvious way. The identity

arrows are the same as in $\mathbf{Bun_0}$. Given an injective function $\rho\colon X \to Y$, the support translation $\rho \cdot c$ is $(\rho X, \mathrm{char}\,\rho^{-1}, \mathrm{root}(\mathrm{id}_{m_0} + \rho^{-1}))$. It is easy to verify that this satisfies the axioms of precategories.

The definitions below recall the construction of the track and the support quotient categories from a well-supported precategory.

Definition 4.4. The ***track*** of $\mathbb{A}$ is a category $\widehat{\mathbb{C}}$ with

- objects: pairs $\langle A, M\rangle$ where $A \in \mathbb{A}$ and $M \in \mathsf{Set}_f$;
- arrows: $\langle A, M\rangle \xrightarrow{f} \langle B, N\rangle$ where $f\colon A \to B$ is in $\mathbb{A}$, $M \subseteq N$ and $|f| = N \backslash M$.

Composition of arrows is as in $\mathbb{A}$. Observe that the definition of $|f|$ ensures that composition is total. We leave it to the reader to check that this defines a category (cf. [12]).

Definition 4.5. The ***support quotient*** of $\mathbb{A}$ is a category $\mathbb{C}$ with

- objects: as in $\mathbb{A}$;
- arrows: equivalence classes of arrows of $\mathbb{A}$, where f and g are equated if there exist a bijective ρ such that $\rho \cdot f = g$.

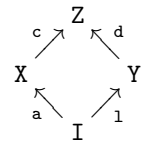
The support quotient is the category of interest, and it is the underlying category of the reactive system under scrutiny.

Example 4.6 (Bunches). The support quotient of **A-Bun** is $\mathbf{Bun}_0$.

There is an obvious functor $F\colon \widehat{\mathbb{C}} \to \mathbb{C}$, the support-quotienting functor. Henceforward we suppose that the precategory $\mathbb{A}$ has a distinguished object I. In the following we use the typewriter font for objects and arrows of $\widehat{\mathbb{C}}$. We make the notational convention that any $\mathtt{A}$ and $\mathtt{f}$ in $\widehat{\mathbb{C}}$ are such that $F(\mathtt{A}) = A$ and $F(\mathtt{f}) = f$.

Definition 4.7 (The LTS). The LTS $\mathbf{FLTS}^c(\mathbb{C})$ has

- States: arrows $a\colon 0 \to n$ in $\mathbb{C}$;
- Transitions: $a \xrightarrow{c} dr$ if and only if there exist $\mathtt{a}, \mathtt{l}, \mathtt{c}, \mathtt{d}$ in $\hat{\mathbb{C}}$ with $\langle F(\mathtt{l}), r\rangle \in R$, $F(\mathtt{d}) \in \mathbb{D}$, and such that



is an IPO.

It is proved in [12] that the support-quotienting functor F satisfies the properties required for the theory of functorial reactive systems [11, 12]. Thus, for instance, if the category $\widehat{\mathbb{C}}$ has enough RPOs, then the bisimulation on $\mathbf{FLTS}^c(\mathbb{C})$ is a congruence.

All the theory presented so far can be elegantly assimilated into the theory of GRPOs. In [12], Leifer predicted instead of precategories, one could consider a bicategorical notion of RPO in a bicategory of supports. This is indeed the case, with GRPOs

being the bicategorical notion of RPO. However, working with ordinals for support sets we can avoid the extra complications bicategories as in the case of **Bun**. It is worth noticing, however, that a bicategory of supports as above and the G-category define below would be biequivalent (cf. [20]).

In the following, we make use of a chosen isomorphism $t_x : x \to \mathrm{ord}(x)$, where for any finite set x, $\mathrm{ord}(x)$ denotes the finite ordinal of the same cardinality. There is an equivalence of categories $F : \mathbf{Set}_f \to \mathbf{Ord}$ which sends x to $\mathrm{ord}(x)$ and, on morphisms, $f : x \to y$ to $t_y f t_x^{-1} : \mathrm{ord}(x) \to \mathrm{ord}(y)$.

Definition 4.8 (G-category of Supports). Given a well-supported precategory $\mathbb{A}$, the G-category of supports $\mathbb{B}$ has

- objects: as in $\mathbb{A}$;
- arrows: $f : A \to B$ where $f : A \to B$ is an arrow of $\mathbb{A}$ and $|f|$ is an ordinal;
- 2-cells: $\rho : f \Rightarrow g$, where ρ is a "structure preserving" support bijection, that is $\rho \cdot f = g$ in $\mathbb{A}$.

Composition is defined as follows. Given $f : A \to B$ and $g : B \to C$,

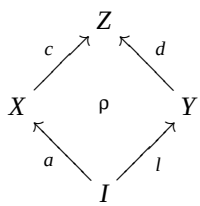
$$g \circ_{\mathbb{B}} f = i_2 \cdot g \circ_{\mathbb{A}} i_1 \cdot f$$

where $|f| \xrightarrow{i_1} |f| \oplus |g| \xleftarrow{i_2} |g|$ is the chosen coproduct diagram in **Ord**.

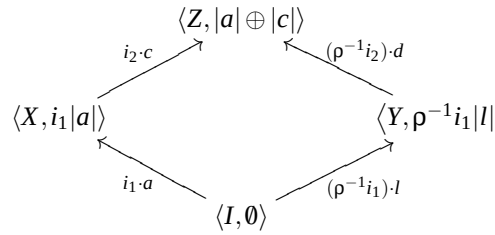
The following theorem guarantees that the LTS generated is the same as the one generated with the more involved theory of functorial reactive systems.

Theorem 4.9. $\mathbf{FLTS}^c(\mathbb{A}) = \mathsf{GTS}(\mathbb{B})$.

Proof. It is enough to present translations between GIPOs in $\mathbb{B}$ and IPOs in $\widehat{\mathbb{C}}$ which preserve the resulting label in the derived LTS. We present the translations, but omit the straightforward proofs. Suppose that *(i)* below

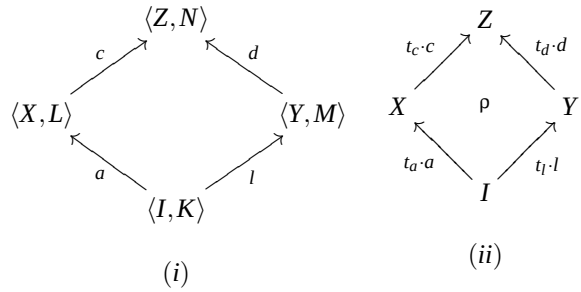


(i)



(ii)

is a GIPO in $\mathbb{B}$. Then we claim that *(ii)* is an IPO in $\mathbb{C}$. Note that *(ii)* is commutative since ρ is a structure-preserving support bijection. Going the other way, suppose that *(i)* below



is an IPO in $\mathbb{C}$. Then *(ii)* is a GIPO in $\mathbb{B}$ where ρ is

$$|t_a \cdot a| \oplus |t_c \cdot c| \xrightarrow{t_a^{-1} \oplus t_c^{-1}} |a| \cup |c| = |l| \cup |d| \xrightarrow{t_l \cup t_d} |t_l \cdot l| \oplus |t_d \cdot d|.$$

□

Example 4.10 (Bunches). The 2-category of supports of the precategory **A-Bun** is **Bun**. Note that a "structure preserving" support bijection is a bunch homomorphism. Indeed, $\rho\colon (X, \mathrm{char}, \mathrm{root}) \to (X', \mathrm{char}', \mathrm{root}')$ if $X' = \rho X$, $\mathrm{char}' = \mathrm{char}\,\rho^{-1}$ and $\mathrm{root}' = \mathrm{root}(\mathrm{id} \oplus \rho^{-1})$ which is the same as saying $\mathrm{char} = \mathrm{char}'\,\rho$ and $\mathrm{root} = \mathrm{root}'(\mathrm{id} \oplus \rho)$.

5 Conclusion

We have extended our theory of GRPOs initiated in previous work in order to strengthen existing techniques for deriving operational congruences for reduction systems in the presence of non trivial structural congruences. In particular, this paper has shown that previous theories can be recast using G-reactive systems and GRPOs at no substantial additional complexity. Also, we proved that the theory is powerful enough to handle the examples considered so far in the literature. Therefore, we believe that it constitutes a natural starting point for future investigations towards a fully comprehensive theory.

It follows from Theorem 4.9 that G-categories are at least as expressive as well-supported precategories. A natural consideration is whether a reverse translation may exist. We believe that this is not the case, as general G-categories appear to carry more information than precategories.

References

[1] M. Bugliesi, S. Crafa, M. Merro, and V. Sassone. Communication interference in mobile boxed ambients. In *Foundations of Software Technology and Theoretical Computer Science, FST&TCS 02*, volume 2556 of *Lecture Notes in Computer Science*, pages 71–84. Springer, 2002.

[2] A. Carboni, S. Lack, and R. F. C. Walters. Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra*, 84(2):145–158, February 1993.

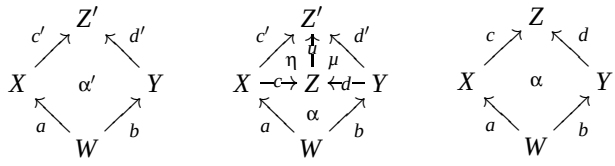
[3] G. Castagna and F. Zappa Nardelli. The seal calculus revised. In *Foundations of Software Technology and Theoretical Computer Science, FST&TCS 02*, volume 2556 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2002.

[4] J. C. Godskesen, T. Hildebrandt, and V. Sassone. A calculus of mobile resources. In *Int. Conf. on Concurrency Theory, CONCUR 02*, volume 2421 of *Lecture Notes in Computer Science*, pages 272–287. Springer, 2002.
[5] M. Hennessy and M. Merro. Bisimulation congruences in safe ambients. In *Principles of Programming Languages, POPL 02*, pages 71–80. ACM Press, 2002.
[6] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
[7] O. H. Jensen and R. Milner. Bigraphs and transitions. In *Principles of Programming Languages, POPL 03*. ACM Press, 2003.
[8] G. M. Kelly. Elementary observations on 2-categorical limits. *Bull. Austral. Math. Soc.*, 39:301–317, 1989.
[9] G. M. Kelly and R. H. Street. Review of the elements of 2-categories. *Lecture Notes in Mathematics*, 420:75–103, 1974.
[10] F. W. Lawvere. Functorial semantics of algebraic theories. *Proceedings, National Academy of Sciences*, 50:869–873, 1963.
[11] J. Leifer. *Operational congruences for reactive systems*. Phd thesis, University of Cambridge, 2001.
[12] J. Leifer. Synthesising labelled transitions and operational congruences in reactive systems, parts 1 and 2. Technical Report RR-4394 and RR-4395, INRIA Rocquencourt, 2002.
[13] J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *Int. Conf. on Concurrency Theory, CONCUR 00*, Lecture Notes in Computer Science, pages 243–258. Springer, 2000.
[14] R. Milner. Calculi for interaction. *Acta Informatica*, 33(8):707–737, 1996.
[15] R. Milner. Bigraphical reactive systems: Basic theory. Technical Report 523, Computer Laboratory, University of Cambridge, 2001.
[16] A. W. Roscoe. *The Theory and Practice of Concurrency*. Prentice Hall, 1997.
[17] V. Sassone and P. Sobociński. Deriving bisimulation congruences: A 2-categorical approach. *Electronic Notes in Theoretical Computer Science*, 68(2), 2002.
[18] V. Sassone and P. Sobociński. Deriving bisimulation congruences: 2-categories vs. precategories. Technical Report RS-03-1, BRICS, January 2003.
[19] P. Sewell. From rewrite rules to bisimulation congruences. *Lecture Notes in Computer Science*, 1466:269–284, 1998.
[20] R. H. Street. Fibrations in bicategories. *Cahiers de topologie et géométrie différentielle*, XXI-2:111–159, 1980.

A Basic Properties of GRPOs

The next two lemmas explain the relationships between GRPOs and GIPOs.

Lemma A.1 (GIPOs from GRPOs). *If $\langle Z, c, d, u, \alpha, \eta, \mu \rangle$ is a GRPO for (i) below, as illustrated in (ii), then (iii) is a GIPO.*

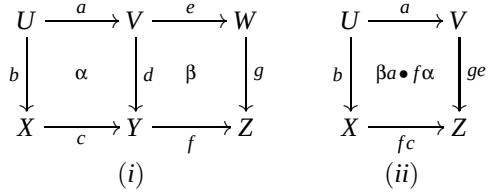


(i) (ii) (iii)

Lemma A.2 (GRPOs from GIPOs). *If square (iii) above is a* GIPO*, (i) has a* GRPO*, and* $\langle Z, c, d, u, \alpha, \eta, \mu \rangle$ *is a candidate for it as shown in (ii), then* $\langle Z, c, d, u, \alpha, \eta, \mu \rangle$ *is a* GRPO *for (i).*

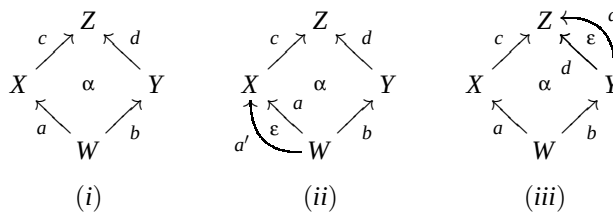
The following lemmas from [17] state the basic properties of GRPOs.

Lemma A.3. Suppose that diagram *(ii)* below has a GRPO.



1. If both squares in *(i)* are GIPOs then the rectangle of *(i)* is a GIPO
2. If the left square and the rectangle of *(i)* are GIPOs then so is the right square.

Lemma A.4. Suppose that diagram *(i)* below is a GIPO.



Then the regions obtained by pasting the 2-cells in *(ii)* and *(iii)* are GIPOs.