



# Single Pushout Rewriting in Comprehensive Systems

Harald König<sup>1,2</sup>(✉)  and Patrick Stünkel<sup>2</sup> 

<sup>1</sup> University of Applied Sciences, FHDW, Hannover, Germany  
harald.koenig@fhdw.de

<sup>2</sup> Høgskulen på Vestlandet, Bergen, Norway  
past@hvl.no

**Abstract.** The elegance of the *single-pushout* (SPO) approach to graph transformations arises from substituting total morphisms by partial ones in the underlying category. Thus, SPO's applicability depends on the durability of pushouts after this transition. There is a wide range of work on the question when pushouts exist in categories with partial morphisms starting with the pioneering work of Löwe and Kennaway and ending with an essential characterisation in terms of an exactness property (for the interplay between pullbacks and pushouts) and an adjointness condition (w.r.t. inverse image functions) by Hayman and Heindel.

Triple graphs and graph diagrams are frameworks to synchronize two or more updatable data sources by means of internal mappings, which identify common sub-structures. *Comprehensive systems* generalise these frameworks, treating the network of data sources and their structural inter-relations as a homogeneous comprehensive artifact, in which partial maps identify commonalities. Although this inherent partiality produces amplified complexity, Heindel's characterisation still yields cocompleteness of the category of comprehensive systems equipped with closed partial morphisms and thus enables computing by SPO graph transformation.

**Keywords:** Single Pushout Rewriting · Partial morphism · Category theory · Hereditary pushout · Upper adjoint · Comprehensive system

## 1 Introduction and Motivation

We want to dedicate this paper to Michael Löwe, the founder of the single-pushout approach [12] and simultaneously a pioneer in the investigation of categories of partial algebras with partial morphisms between them [13].

In this paper, we want to combine these two theories. We introduce the category of *comprehensive systems*, formally a category in which the inner structure of the objects can be described with partial maps, and will show that SPO rewriting is applicable in this category.

---

In Honour of Michael Löwe, 1956 - 2019.

Comprehensive Systems have been introduced in [21] (see also [22]) as a means for global consistency management, representing a collection of inter-related systems. To provide an intuition of a comprehensive system (Definition 3 in Sect. 3), take a look at Fig. 1. There are three conceptual models  $A_1, A_2, A_3$ , which depict persons (👤) with certain features: In  $A_1$  a phone number (☎️) is assigned to the person, in  $A_2$  and  $A_3$  the person possesses a home address (🏠), and in  $A_3$  persons additionally may have a business address (🏢). You may think of  $A_{1/2/3}$  representing (excerpts of) the contents of three databases possibly in three distributed IT-systems, the first stores records of persons and phone numbers while the second and third store records of persons and addresses.

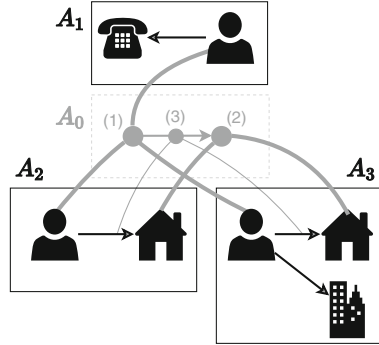


Fig. 1. Comprehensive system

In many cases it is necessary to maintain global consistency of the databases’ contents, especially in the presence of inter-model constraints [20]. Let us assume the following constraint:

**IMC** Every person with a business address must either provide a phone number or a home address or both.

Violations of this constraint can only be discovered, if common elements in the models are identified. Thus one has to specify that a recorded person in  $A_1$  is actually *the same* as some person record in  $A_2$  and/or  $A_3$ . Such commonality specifications extend the modelling language and are commonly used in practice, e.g. [5]. In Fig. 1 we employ grey-coloured “tentacles” to visualise three commonalities: (1) The three person records, (2) the two home addresses, and also (3) the two assignments of the home address to the person.

Note that  $A_{1/2/3}$  formally represent directed graphs. But the junctions of each commonality (•) – called *commonality representatives* – form another graph  $A_0$  in Fig. 1. Elements of  $A_0$  witness common elements among  $A_{1/2/3}$  and these commonalities must obviously respect node-edge-incidences (see the person to home address assignment), such that their respective outgoing grey lines are in fact graph morphisms  $a_i : A_0 \rightarrow A_i, i \in I = \{1, 2, 3\}$ .

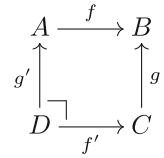
For  $|I| = 2$  the underlying star-shape of comprehensive systems (finite collections of arrows  $(a_i)_{i \in \{1, \dots, n\}}$  with common source) reduces to the span shape  $\bullet \leftarrow \bullet \rightarrow \bullet$ , which is the underlying setting for triple graphs [18], the common source in the middle specifying the commonality graph. An extension of triple graphs are graph diagrams [23], a framework for *multi-ary* model synchronisation. Since multi-ary commonality relations such as the ternary tentacles of identical person records in Fig. 1 can not be encoded with several binary relations [19], one must distinguish relations of different arity in the underlying shape for graph diagrams: E.g. in Fig. 1, a shape with two nodes is required: One node

specifies the existence of a graph containing binary home address commonalities and one node is used for ternary commonalities of person records. In larger system landscapes ( $n > 3$ ), there may be many more commonality relations of arbitrary arity  $k \leq n$ , which would cause a considerable amount of heterogeneity of commonality nodes in the underlying shape for graph diagrams. Moreover, this schema and hence the basic setting for implementations must be altered, whenever new commonality relations are added.

We showed in [21, 22] that comprehensive systems are a homogeneous generalisation of graph diagrams. They are *homogeneous*, because we need only one node to cope with commonality relations of arbitrary arity (the center of the star-shape specifying commonality graph  $A_0$ ) and must not alter the base setting, if new relations are added. It is a *generalisation*, because we can implement each graph diagram as a comprehensive system, i.e. we can jointly collect different commonalities into one graph  $A_0$ .

An important distinction, however, is that graph morphisms  $a_i : A_0 \rightarrow A_i$  in comprehensive systems are allowed to be *partial*. E.g.  $a_1 : A_0 \rightarrow A_1$  in Fig. 1 is undefined on (2) and (3) in  $A_0$ . Nevertheless can we show that comprehensive systems form a category  $\mathbb{CS}$ , in which graph rewriting, especially SPO rewriting, is possible. For this, we will also consider the category  $\text{Par}(\mathbb{CS})$ , i.e.  $\mathbb{CS}$  equipped with *partial* morphisms, cf. Sect. 2. Although this requires handling both *intrinsic* and *extrinsic* partiality, we can prove existence of all pushouts in this category (Theorem 1 in Sect. 4) and hence demonstrate applicability of SPO rewriting.

We expect the reader to have basic knowledge in category theory. For categorical artefacts, we employ the following notations: *Categories* like  $\mathbb{C}$  will be denoted in a double-struck font. When distinguishing between members of  $\mathbb{C}$ , we write  $|\mathbb{C}|$  (or just  $\mathbb{C}$ ) for its objects and  $\text{Arr}(\mathbb{C})$  for its morphisms. Moreover, there are identities  $\text{id}_A : A \rightarrow A$  and composition, e.g.  $g \circ f$ , for  $f : A \rightarrow B$  and  $g : B \rightarrow C$ .  $\mathbb{SET}$  is the category of sets and total mappings and we will use the letter  $\mathbb{G}$  for categories that are based on a signature with unary operation symbols only, see Sect. 3 for more details. Monomorphisms ( $\rightarrow$ ), epimorphisms ( $\twoheadrightarrow$ ) and – if applicable – inclusions ( $\hookrightarrow$ ) have special notations. We furthermore expect the reader to be familiar with basic *universal constructions* like pullbacks, coproducts, and pushouts. When describing a pullback as in Fig. 2, we either say that the span  $(g', f')$  is the pullback of co-span  $(f, g)$  or we call  $g'$  the pullback of  $g$  along  $f$ . Moreover, the pullback object  $D$  in this construction is highlighted with a small adjacent right angle.



**Fig. 2.** Pullback along  $f$

## 2 SPO Rewriting

### 2.1 Graph-Like Structures

A category  $\mathbb{G}$  is called *based on a signature with unary operation symbols only*, if it is isomorphic to a category of total algebras w.r.t. a signature, which only contains sorts and unary operation symbols. The simplest example in our context –

and the rationale behind using letter  $\mathbb{G}$  – are directed graphs, which are based on a signature with sorts  $E$  and  $V$  and two unary operation symbols  $s, t : E \rightarrow V$ . We do not endorse directed graphs in particular and could likewise choose  $\mathbb{G}$  to be given by E-Graphs [4] or bipartite artefacts like condition-event-nets. It is well-known that all these categories are topoi and thus possess all limits (e.g. pullbacks) and colimits (coproducts, pushouts) [6].

Objects of such categories  $\mathbb{G}$  are sometimes called “graph-like structures”, as e.g. in [14], and thus we will simply call  $\mathbb{G}$ -objects and morphisms “graphs” and “graph morphisms” bearing in mind the above mentioned more general setting.

*Remark:*  $\mathbb{G}$  will serve as the base category (or base structure) for assembling comprehensive systems, cf. Fig. 1. Actually, we could have traded  $\mathbb{G}$  for more general (weak) adhesive (HLR) categories w.r.t. an admissible subclass  $\mathcal{M}$  of all monomorphisms.<sup>1</sup> Adhesive HLR categories have mainly been introduced to model *attributes*, which poses some challenges regarding adhesiveness, in turn requiring to work with special subclasses of morphisms, which are isomorphic on the “data part”. However, we restrict ourselves to graph-like structures because we are not focusing on attributes here and we want to stay in the tradition of Michael Löwe, who originally investigated graph-like structures only. Since graph-like structures are sufficiently concrete, we can actually refer to an element  $x \in A$  for some  $A \in \mathbb{G}$ , i.e. an element of some carrier set of graph  $A$ . Likewise, “ $\forall x \in A$ ” means “for all  $x$  of any sort  $s$  in the carrier set of  $A$ ”.

## 2.2 Partial Map Categories

Consider an arbitrary category  $\mathbb{C}$  with pullbacks. Michael had the courage to leave the comfortable world of *total*  $\mathbb{C}$ -morphisms and utilised *partial* morphisms [17] for the SPO approach. While other researchers adhered to total morphisms, he forcefully followed through with partiality and proved that it is worthwhile [12]. He used the following definitions: A  $\mathbb{C}$ -span  $X \xleftarrow{m} \text{dom}(f) \xrightarrow{f} Y$ , is

equivalent to a second span  $X \xleftarrow{m'} \text{dom}(f') \xrightarrow{f'} Y$ , if and only if there is an isomorphism  $i : \text{dom}(f) \rightarrow \text{dom}(f')$  such that  $m' \circ i = m$  and  $f' \circ i = f$ . A *partial morphism* is an equivalence class w.r.t. this relation, denoted by  $\langle m, f \rangle : X \rightarrow Y$ , i.e. the pair  $(m, f)$  is a representative of its equivalence class.  $\langle m, f \rangle$  is called *total*, if  $m$  is an isomorphism, and we use the usual arrow tip in this case:  $\langle m, f \rangle : X \rightarrow Y$ .

It is then easy to see that the objects of  $\mathbb{C}$  together with partial morphisms constitute a new category  $\text{Par}(\mathbb{C})$ : An identity is the equivalence class of the identity span and composition of  $\langle m_1, f_1 \rangle : G_1 \rightarrow G_2$  and  $\langle m_2, f_2 \rangle : G_2 \rightarrow G_3$  is given by constructing the pullback span  $(m'_2, f'_1)$  of co-span  $(f_1, m_2)$  yielding the composed partial morphism  $\langle m_1 \circ m'_2, f_2 \circ f'_1 \rangle$ . This is in fact independent of the choice of pullbacks and independent of the choice of representatives.

<sup>1</sup> It seems that the subsequent proofs can still be carried out, if  $\mathbb{G}$  is such a more general structure. We will provide respective facts from time to time.

Furthermore we obtain an identity-on-objects functor

$$\Gamma : \begin{cases} \mathbb{C} \rightarrow \mathbf{Par}(\mathbb{C}) \\ (A \xrightarrow{f} B) \mapsto (A \xrightarrow{\langle id_A, f \rangle} B) \end{cases}$$

called the *graphing functor* [8], i.e. a canonical embedding of totality into partiality.

**Definition 1 (Hereditary Pushout).** [8] *Any pushout in  $\mathbb{C}$  is called hereditary, if its  $\Gamma$ -image is a pushout in  $\mathbf{Par}(\mathbb{C})$ . If all pushouts exist in  $\mathbb{C}$  and they are all hereditary, we say that  $\mathbb{C}$  is a hereditary pushout category.*

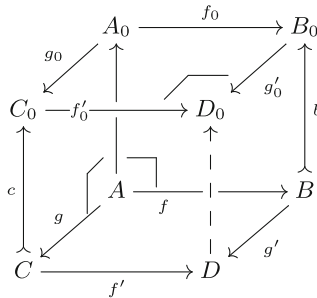
The following result can be found in [10]:

**Proposition 1.** *If the  $\Gamma$ -image of a  $\mathbb{C}$ -span has a pushout in  $\mathbf{Par}(\mathbb{C})$ , then this cocone consists of two total morphisms, which are the  $\Gamma$ -image of the pushout of this span in  $\mathbb{C}$ .  $\square$*

For  $\mathbb{C} := \mathbb{G}$ , we can refer to elements inside objects of  $\mathbb{G}$ , such that we will work with representatives  $G \xleftarrow{m} \text{dom}(f) \xrightarrow{f} H$  of a partial morphism, in which the left leg  $m$  is chosen as the effective inclusion of  $\text{dom}(f)$ , the domain of definition of the partial morphism, into  $G$ . Since the name  $m$  is of minor importance, we may as well write  $f : G \rightarrow H$ . In this setting, we will call  $f$  “total”, if the inclusion  $m$  is the identity. For the remainder of this paper we will use  $\mathbb{G}$ -inclusions when there is a choice for monomorphisms (replacing  $\rightarrow$  with  $\hookrightarrow$ ).

The following result was stated in [7] but fully worked out already in [12]:

**Proposition 2.**  *$\mathbb{G}$  is a hereditary pushout category.  $\square$*



**Fig. 3.** Commutative cube

Finally, hereditariness can equivalently be characterised as follows:

**Proposition 3 (Equivalent Characterisation of Hereditariness).** [9] *A pushout like the top face in the cube in Fig. 3 is hereditary, if and only if in any commutative cube as in Fig. 3 with rear faces being pullbacks and vertical*

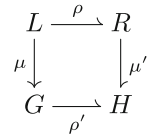
front left and back right arrows ( $c$  and  $b$  in Fig. 3) being monomorphisms, the following equivalence holds: The bottom face is a pushout if and only if (1) the two front faces are pullbacks and (2) the vertical front right arrow (the dashed arrow in Fig. 3) is a monomorphism.  $\square$

### 2.3 Rewriting Rule and Derivation

While the *double-pushout approach* (DPO) [4] requires the construction of a pushout complement and another pushout in the underlying category  $\mathbb{C}$ , the introduction of partial morphisms enables graph transformations to be expressed by a *single pushout* in  $\text{Par}(\mathbb{C})$ . This elegant alternative to DPO was initiated by Raoult [16] and then fully worked out in Michael’s PhD Thesis [12].

**Definition 2 (Rule, Match, Derivation, Conflict-Freeness).** *An SPO rule is a morphism  $L \xrightarrow{\rho} R$  of  $\text{Par}(\mathbb{C})$ . A match for  $\rho$  at (host)  $G \in \mathbb{C}$  is a total morphism  $\mu : L \rightarrow G$ . A pushout of  $\rho$  and  $\mu$  in  $\text{Par}(\mathbb{C})$  generates the (SPO-) derivation  $G \xrightarrow{\rho, \mu} H$  with trace  $\rho'$  and co-match  $\mu'$ , see Fig. 4. The match  $\mu$  is called conflict-free, if  $\mu'$  is a total morphism.*

Computing by SPO graph transformation requires the existence of pushouts in  $\text{Par}(\mathbb{C})$  and it requires conflict freeness of matches, cf. Definition 2, if one wants to avoid partial matches. We can prove existence of pushouts for  $\text{Par}(\text{CS})$  in Sect. 4, but - in contrast to a simple criterion for conflict freeness in terms of injectivity of  $\mu$  on delete-preserve pairs [12] - we have to leave the question of a criterion for conflict freeness in  $\text{Par}(\text{CS})$  for future research.



**Fig. 4.** SPO derivation

## 3 Comprehensive Systems

For now and the rest of the paper, we fix a sufficiently large number  $n \in \mathbb{N}$ .<sup>2</sup>

### 3.1 Definitions and Background

**Definition 3 (Comprehensive System).** *Let  $(A_i)_{0 \leq i \leq n}$  be an  $n+1$ -tuple of  $\mathbb{G}$ -objects. We call*

- $(A_j)_{1 \leq j \leq n}$  the Components and
- $A_0$  the Commonality Representatives

of a Comprehensive System

$$\mathbf{A} := (a_j : A_0 \rightarrow A_j)_{j \in \{1, \dots, n\}}$$

*i.e. of an  $n$ -tuple of partial graph morphisms  $(a_j)_{1 \leq j \leq n}$ , which we call projections<sup>3</sup>.*

<sup>2</sup> Usually the number of distributed systems under consideration.

<sup>3</sup> One might consider elements of  $A_0$  to be vectors (of arbitrary arity), in which common elements are listed, hence the term “projection” for the  $a_j$ .

In order to make reading easier, we always use letter  $i$ , if indexing comprises graphs  $A_0, A_1, \dots, A_n$  and we use letter  $j$ , if indexing is only over the components  $A_1, \dots, A_n$ . Moreover, we denote comprehensive systems with bold face letters.

Comprehensive systems admit an all-embracing view on a system of possibly heterogeneously typed components, in which all necessary informational overlaps are coded, cf. Fig. 1. They have been treated on the level of graphs in [20] and – on a more abstract level – in [3].

**Definition 4 (Morphism of Comprehensive Systems).** *Let  $\mathbf{A} := (a_j : A_0 \rightarrow A_j)_{j \in \{1, \dots, n\}}$  and  $\mathbf{A}' := (a'_j : A'_0 \rightarrow A'_j)_{j \in \{1, \dots, n\}}$  be two comprehensive systems. A morphism  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{A}'$  is a family  $(f_i : A_i \rightarrow A'_i)_{0 \leq i \leq n}$  of total  $\mathbb{G}$ -morphisms, such that for all  $1 \leq j \leq n$*

$$a'_j \circ f_0 = f_j \circ a_j \quad (1)$$

*holds in  $\text{Par}(\mathbb{G})$ , the category of graphs and partial morphisms.*

Whenever we mention morphisms  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{A}'$  between comprehensive systems, we implicitly assume the components of  $\mathbf{A}$  and  $\mathbf{A}'$  be denoted as in Definition 3 and we assume the constituents of  $\mathbf{f}$  be denoted as in Definition 4.

There is the obvious identical morphism  $\text{id}_{\mathbf{A}}$  for each comprehensive system  $\mathbf{A}$  and composition can be defined componentwise. Hence we obtain

**Proposition 4 (Category  $\mathbb{CS}$  and Component Functors).** *Let  $\mathbb{G}$  be a category as described above.*

- *Comprehensive Systems and morphisms between them constitute a category, denoted  $\mathbb{CS}_{\mathbb{G}}$  or often just  $\mathbb{CS}$ , if the base category is clear from the context.*
- *For each  $i \in \{0, \dots, n\}$  there is the component functor  $C_i : \mathbb{CS} \rightarrow \mathbb{G}$  defined by  $C_i(\mathbf{f} : \mathbf{A} \rightarrow \mathbf{A}') = A_i \xrightarrow{f_i} A'_i$  for any  $\mathbf{f}$  defined as in Definition 4.  $\square$*

It is important to note that we claim (1) to hold in  $\text{Par}(\mathbb{G})$  and not in  $\mathbb{G}!$  Let's investigate the consequences: Recall that the definition of composition of partial morphisms involves pullbacks. In the situation in Fig. 5 this enforces that a pullback of  $\subseteq'_j$  along  $f_0$  (to express the composition of  $\langle \subseteq'_j, a'_j \rangle$  and  $\langle \text{id}_{A_0}, f_0 \rangle$ ) can be chosen to be equal to  $\subseteq_j$ , the inclusion of the domain of definition of  $f_j \circ a_j$  into  $A_0$ , see the upper square in Fig. 5, and it enforces that the lower square commutes. If  $x \in A_0$ , this observation is equivalent to the statement  $x \in \text{dom}(a_j) \iff f_0(x) \in \text{dom}(a'_j)$ , because “ $\implies$ ” corresponds to commutativity of the upper square and “ $\impliedby$ ” corresponds to the pullback property (in  $\text{SET}$  and hence in  $\mathbb{G}$ , in which pullbacks are constructed sortwise). This yields a more handy admissibility characterisation for  $\mathbb{CS}$ -morphisms:

**Proposition 5 (Preservation and Reflection of Definedness).**  *$\mathbf{f} : \mathbf{A} \rightarrow \mathbf{A}'$  as defined in Definition 4 is a morphism of comprehensive systems if and only if for all  $j \in \{1, \dots, n\}$  and for all  $x \in A_0$ :*

$$a_j(x) \text{ is defined} \iff a'_j(f_0(x)) \text{ is defined} \quad (2)$$

*and the lower square in Fig. 5 commutes.  $\square$*

Usually a morphism between two partial algebras  $A$  and  $B$  requires only preservation of definedness, i.e. “ $\Rightarrow$ ” in (2). In the next section, we justify why we additionally need reflection of definedness.

### 3.2 Why Must Definedness Be Reflected?

Since our goal is to show that SPO rewriting is applicable for comprehensive systems, we must show that  $\text{Par}(\text{CS})$  possesses all pushouts.

Assume we would not have claimed reflection of definedness for  $\text{CS}$ -morphisms, but only commutativity of the two squares in Fig. 5, which is equivalent to claiming the properties of Proposition 5 except for the implication direction “ $\Leftarrow$ ” in (2). In this case let’s consider for  $n = 1$  two simple comprehensive systems. Let  $\mathbb{G} = \text{SET}$  and  $A_0 = \{*\}$  and  $A_1 = \{\bullet\}$  be two one-element sets and let  $\mathbf{A} = (a_1 : A_0 \rightarrow A_1)$  with  $a_1$  the totally undefined map depicted with  $(* \ \bullet)$  and  $\mathbf{A}' = (a'_1 : A_0 \rightarrow A_1)$  with  $a'_1$  the unique total map from  $A_0$  to  $A_1$  depicted  $(* \mapsto \bullet)$ . If we only work with preservation of definedness, then morphism  $\langle id_A, f \rangle : \mathbf{A} \rightarrow \mathbf{A}'$ , in which  $f_0$  maps  $*$  to  $*$  and  $f_1$  maps  $\bullet$  to  $\bullet$ , is an admissible morphism. We claim that the span  $\mathbf{A}' \xleftarrow{\langle id_A, f \rangle} \mathbf{A} \xrightarrow{\langle id_A, f \rangle} \mathbf{A}'$  does not possess a pushout in  $\text{Par}(\text{CS})$ .

If there would be a pushout of this span of two total morphisms in  $\text{Par}(\text{CS})$ , then, by Proposition 1, it must coincide with the pushout of them in  $\text{CS}$ . Since  $f$  is an epimorphism in  $\text{CS}$  (because all  $f_j$  are epimorphisms in  $\mathbb{G}$ ), the pushout in  $\text{CS}$  must have  $p_1 = p_2 = id_{A'}$  as cocone, see the left top square in Fig. 6. The two partial morphisms  $\langle m, h \rangle$  and  $\langle id_{A'}, id_{A'} \rangle$  let the outer rectangle of partial morphisms commute, i.e.

$$\langle m, h \rangle \circ \langle id_A, f \rangle = \langle id_{A'}, id_{A'} \rangle \circ \langle id_A, f \rangle$$

in  $\text{Par}(\text{CS})$ , because the pullback object of  $m$  and  $f$  equals the pullback object of  $id_{A'}$  and  $f$  in  $\text{CS}$ . If there would be a unique mediator  $u$ , see the dashed line in the diagram, we must have  $u = \langle id_{A'}, id_{A'} \rangle$ , because the lower rhombus must be commutative. However, for this  $u$  the right rhombus fails to be commutative, because  $u \circ \langle id_{A'}, p_2 \rangle = \langle id_{A'}, id_{A'} \rangle \neq \langle m, h \rangle$ .

This example shows that we cannot expect to have all pushouts in  $\text{Par}(\text{CS})$ , if we would not require reflection of definedness. And this is true, even if the two morphisms, for which the pushout shall be constructed, are total monomorphisms.

### 3.3 Important Properties

Let’s now assume all  $\text{CS}$ -morphisms to reflect definedness.

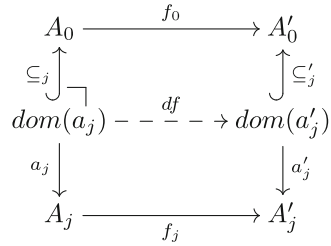


Fig. 5. Composing partial and total morphisms

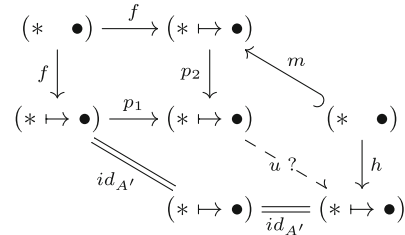


Fig. 6. Counterexample



In the sequel, we will use formulations like “a property is valid *componentwise*” in  $\mathbb{CS}$  or some construction “is carried out *componentwise*”. Since many of the following considerations are based on this methodology, we give a formalisation: “Pushout”, “Pullback”, “Monomorphism”, “Commutativity” impose truth of a predicate (a certain property) on a diagram in a category  $\mathbb{C}$ . For pushouts and pullbacks the underlying diagram is a square, for the predicate “Monomorphism” it is a single arrow, for “Commutativity” it is an appropriate triangle of arrows. E.g.  $\mathbb{CS}$ -morphism  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{B}$  is a componentwise monomorphism means that each  $f_i$  is a  $\mathbb{G}$ -monomorphism. More precisely: Given a diagram  $\mathcal{D}$  of any of the above mentioned shapes in  $\mathbb{CS}$ , let  $\mathcal{D}_i := \mathcal{C}_i(\mathcal{D})$  with component functor  $\mathcal{C}_i$  from Proposition 4, then the predicate  $p$  is true componentwise if and only if it is true for  $\mathcal{D}_i$  in  $\mathbb{G}$  for all  $i \in \{0, \dots, n\}$ .

Another formulation is “*componentwise construction of predicate  $p$* ”, where  $p$  is based on a certain universal property. If e.g.  $p$  is the predicate for pushouts, componentwise construction of a  $\mathbb{CS}$ -cospan  $\mathbf{C} \xrightarrow{\mathbf{f}'} \mathbf{D} \xleftarrow{\mathbf{g}'} \mathbf{B}$  from a  $\mathbb{CS}$ -span  $\mathbf{C} \xleftarrow{\mathbf{g}} \mathbf{A} \xrightarrow{\mathbf{f}} \mathbf{B}$  consists of two steps: In a first step one constructs pushout cospans  $\mathcal{C}_i \xrightarrow{f'_i} \mathcal{D}_i \xleftarrow{g'_i} \mathcal{B}_i$  of spans  $\mathcal{C}_i(\mathbf{C} \xleftarrow{\mathbf{g}} \mathbf{A} \xrightarrow{\mathbf{f}} \mathbf{B})$  for each  $i \in \{0, \dots, n\}$ . In a second step one tries to define the projections  $d_j$  in  $\mathbf{D} := (d_j : D_0 \rightarrow D_j)_{1 \leq j \leq n}$ , cf. Definition 3, with the help of the pushouts’ unique mediators. The cospan morphisms  $\mathbf{f}'$  and  $\mathbf{g}'$  consist of the respective components  $(f'_i)_{0 \leq i \leq n}$  and  $(g'_i)_{0 \leq i \leq n}$ . The phrase “ *$p$  can be constructed componentwise*” then means that the newly constructed object  $\mathbf{D}$  is an admissible object according to Definition 3, that the newly created morphisms  $\mathbf{f}'$  and  $\mathbf{g}'$  are admissible according to Definition 4, and that predicate  $p$  holds on the resulting diagram in  $\mathbb{CS}$ , i.e. the square that arises from enhancing the above  $\mathbb{CS}$ -span by the  $\mathbb{CS}$ -cospan yields a pushout in  $\mathbb{CS}$ . Of course, this procedure applies to other universal constructions in a similar way and after such a construction, we know that property  $p$  is valid componentwise.

“Commutativity” is valid componentwise by definition, but we also obtain.

**Proposition 6 (Componentwise Properties of  $\mathbb{CS}$ ).** *Morphism  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{B}$  is a monomorphism if and only if it is such componentwise.*

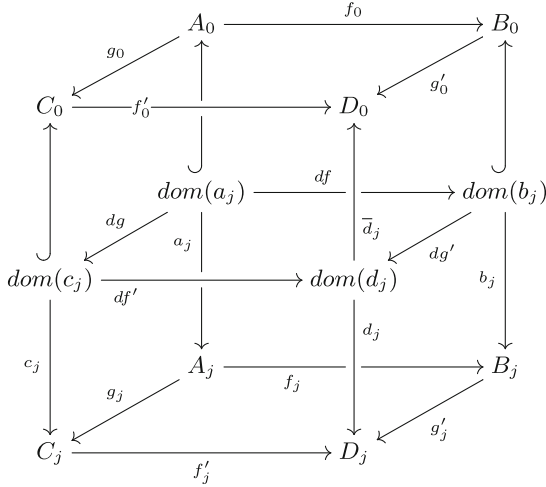
*Moreover,  $\mathbb{CS}$  has*

1. *all pullbacks*
2. *all pushouts*
3. *all coproducts*

*(and is thus cocomplete) and they are constructed componentwise, resp.*

*Proof.* Componentwise validity of monomorphy and componentwise construction of pullbacks have been proved in [11] for so-called  $S$ -cartesian functor categories. We showed in [21] (see also [22]) that - for a certain schema category - this functor category is equivalent to  $\mathbb{CS}$ .

Thus, it remains to prove 2 and 3. For the proof of 2, let a span  $(\mathbf{f} : \mathbf{A} \rightarrow \mathbf{B}, \mathbf{g} : \mathbf{A} \rightarrow \mathbf{C})$  of  $\mathbb{CS}$ -morphisms be given where  $\mathbf{f} = (f_i : A_i \rightarrow B_i)_{0 \leq i \leq n}$  and



**Fig. 7.** Two commutative cubes

$\mathbf{g} = (g_i : A_i \rightarrow B_i)_{0 \leq i \leq n}$ . Resolving these two morphisms into a triple of  $\mathbb{G}$ -morphisms for each  $j \in \{1, \dots, n\}$  as in Fig. 5 and constructing pushouts componentwise in  $\mathbb{G}$ , i.e. for  $f_0$  and  $g_0$ ,  $f_j$  and  $g_j$ , and for the span of resulting domain mappings  $df$  and  $dg$ , see the dashed arrow in Fig. 5, yields two cubes on top of each other for each  $j \in \{1, \dots, n\}$ , cf. Fig. 7, in which the vertical front right arrows  $d_j$  and  $\bar{d}_j$  are unique mediators w.r.t. the middle pushout. Because  $\mathbb{G}$  is a hereditary pushout category by Proposition 2 and because the top face in the upper cube in Fig. 7 is a  $\mathbb{G}$ -pushout and the two back faces are pullbacks (cf. Definition 4 and the definition of composition of partial arrows on page 4), the prerequisite of the equivalent characterization of hereditaryness in Proposition 3 are fulfilled. Hence the fact that the middle layer in Fig. 7 is also a pushout (by construction) implies that the two upper front faces become pullbacks and the vertical upward arrow  $\bar{d}_j$  in the front right can be chosen to be an inclusion arrow. This shows that the componentwise construction indeed yields an admissible comprehensive system

$$\mathbf{D} := (D_0 \xrightarrow{d_j} D_j)_{1 \leq j \leq n}$$

and a commutative square  $(\mathbf{f} : \mathbf{A} \rightarrow \mathbf{B}, \mathbf{g}' : \mathbf{B} \rightarrow \mathbf{D}, \mathbf{g} : \mathbf{A} \rightarrow \mathbf{C}, \mathbf{f}' : \mathbf{C} \rightarrow \mathbf{D})$  in  $\mathbb{CS}$ . It remains to show that it is also a pushout.

Let for this a  $\mathbb{CS}$ -object  $\mathbf{Z} := (Z_0 \xrightarrow{z_j} Z_j)_{1 \leq j \leq n}$  and two  $\mathbb{CS}$ -morphisms  $\mathbf{h} : \mathbf{B} \rightarrow \mathbf{Z}$  and  $\mathbf{k} : \mathbf{C} \rightarrow \mathbf{Z}$  be given such that  $\mathbf{h} \circ \mathbf{f} = \mathbf{k} \circ \mathbf{g}$ . Then componentwise considerations easily yield unique  $\mathbf{u} := (u_i : D_i \rightarrow Z_i)_{0 \leq i \leq n}$  factoring through the components of  $\mathbf{h}$  and  $\mathbf{k}$ , resp., see Fig. 8, which shows the situation involving  $\mathbf{h}$  only. It is easy to see that universality of  $d_j$  and  $\bar{d}_j$  yield commutativity of all squares in Fig. 8, such that it remains to show that  $\mathbf{u}$  is a  $\mathbb{CS}$ -morphism. For this we use the equivalent characterization in Proposition 5, in particular we have to

show (2) for  $\mathbf{u}$ . Let for this  $x \in D_0$  be given. It is well known that pushouts in  $\mathbb{G}$  yield jointly surjective cospans, i.e.  $x$  has a preimage  $y$  in  $C_0$  or in  $B_0$ , cf. again Fig. 7. Assume w.l.o.g. that there is  $y \in B_0$  and  $g'_0(y) = x$  (the case, where there is a preimage in  $C_0$ , is similar). Then again using Proposition 5 several times yields

$$\begin{aligned}
 d_j(x) \text{ is defined} &\iff b_j(y) \text{ is defined} && (\text{because } \mathbf{g}' : \mathbf{B} \rightarrow \mathbf{D} \in \text{Arr}(\mathbb{CS})) \\
 &\iff z_j(h_0(y)) \text{ is defined} && (\mathbf{h} : \mathbf{B} \rightarrow \mathbf{Z} \in \text{Arr}(\mathbb{CS}), \text{ see Fig. 8}) \\
 &\iff z_j(u_0(x)) \text{ is defined} && (h_0 = u_0 \circ g'_0 \text{ and } x = g'_0(y)),
 \end{aligned}$$

which shows that  $\mathbf{u}$  is a  $\mathbb{CS}$ -morphism.<sup>4</sup>

The proof of the existence of coproducts is similar: Let  $(\mathbf{A}^k := (A_0^k \xrightarrow{a_j^k} A_1^k)_{1 \leq j \leq n})_{k \in I}$  be a family of comprehensive systems indexed over some (possibly infinite) index set  $I$ . It is then easy to see that

$$\mathbf{A} := \left( \coprod_{k \in I} A_0^k \xrightarrow{\coprod_{k \in I} a_j^k} \coprod_{k \in I} A_1^k \right)_{1 \leq j \leq n}$$

is the coproduct of them, where  $\coprod_{k \in I} A_i^k$  denotes  $\mathbb{G}$ -coproducts (hence the  $\mathbb{CS}$ -coproduct is constructed componentwise).

For each  $j$  the partial morphism  $\coprod_{k \in I} a_j^k$  is defined to be equal to  $a_j^r$  on each  $A_0^r$ . The unique mediator for a family  $(\mathbf{f}^k : \mathbf{A}^k \rightarrow \mathbf{B})$  can be shown to be a  $\mathbb{CS}$ -morphism by similar arguments as above for  $\mathbf{u}$ . It is well-known that all colimits can be constructed from binary pushouts and coproducts [1], hence  $\mathbb{CS}$  is indeed cocomplete.  $\square$

The equivalent characterization of hereditaryness in Proposition 3 uses the predicates pushout, pullback, monomorphism, and commutativity, of which we have shown that validity in  $\mathbb{CS}$  is equivalent to componentwise validity. By jumping back and forth from a comprehensive system to its components, this yields

**Corollary 1.**  $\mathbb{CS}$  is a hereditary pushout category.  $\square$

Although it is not the focus of this paper, we mention another important consequence for the application of graph transformations in  $\mathbb{CS}$ :

**Corollary 2.**  $\mathbb{CS}$  is a weak adhesive HLR category [4] w.r.t. the class of all monomorphisms.

*Proof.* Heindel proves in [8], Prop. 8.1 that this conclusion can be drawn from Corollary 1, if pushouts are always stable under pullbacks, i.e. the implication

<sup>4</sup> A more general proof has been given in [11], if  $\mathbb{G}$  is a (variant of an) adhesive category, such that the result carries over to these base structures, as well.

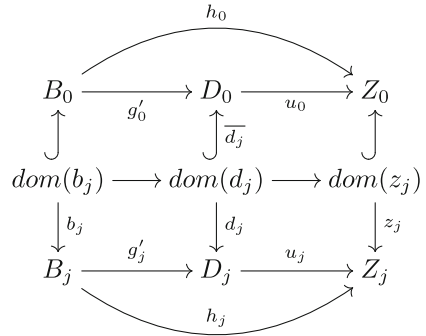


Fig. 8. Admissibility of  $u$

“top face pushout, all side faces pullbacks  $\Rightarrow$  bottom face pushout” holds for all choices of vertical morphisms in Fig. 3. But this implication is true in  $\mathbb{CS}$  by Proposition 6 and because this holds in  $\mathbb{G}$  [6].  $\square$

This corollary guarantees validity of the classical theorems for DPO rewriting such as Local Church Rosser, Parallelism, or Local Confluence Theorem to hold in  $\mathbb{CS}$ , as well.<sup>5</sup>

## 4 The Partial Map Category of Comprehensive Systems Admits All Pushouts

The goal of this section is to prove that SPO rewriting is well possible for comprehensive systems  $\mathbb{CS}$  by showing that the category  $\text{Par}(\mathbb{CS})$  possesses all pushouts. This will follow mainly from a result of Hayman and Heindel:

**Proposition 7 (Existence of Pushouts of Partial Maps, [7]).** *Let  $\mathbb{C}$  be a category with pullbacks in which for each span  $C \xleftarrow{g} A \xrightarrow{f} B$  of morphisms there is a cospan  $C \xrightarrow{f'} D \xleftarrow{g'} B$  making the resulting square commutative.  $\text{Par}(\mathbb{C})$  has all pushouts if and only if  $\mathbb{C}$  is a hereditary pushout category and inverse image functions have upper adjoints.*  $\square$

Section 4.1 is devoted to define inverse image functions and upper adjoints and Sect. 4.2 will show that they exist in  $\mathbb{CS}$ .

### 4.1 Upper Adjoints in General ...

Let in a category  $\mathbb{C}$  with pullbacks an object  $A$  be given. There is the semilattice  $\text{Sub}(A)$  of subobjects of  $A$ , which consists of all monomorphisms  $m : M \rightarrowtail A$  modulo the equivalence relation  $m \equiv (m' : M' \rightarrowtail A)$ , where  $m \equiv m'$ , if and only if there is an isomorphism  $i : M \rightarrow M'$  such that  $m' \circ i = m$ . In the sequel, subobjects will be denoted by small letters  $m, n, \dots$  and we implicitly assume their domain to be the corresponding upper case letter  $M, N, \dots$

The following definitions are well-known from the literature, e.g. [6]. We write  $m \sqsubseteq m'$ , if there is a (necessarily unique) morphism  $f : M \rightarrow M'$  with  $m' \circ f = m$ . In such a way  $\sqsubseteq$  becomes a partial order and in fact a semilattice with meets, because  $\mathbb{C}$  has pullbacks and the meet of  $m$  and  $m'$  is the diagonal in the pullback of  $m$  and  $m'$ . Furthermore one says that  $\mathbb{C}$  has images, if for each  $\mathbb{C}$ -morphism  $f : X \rightarrow A$  there is a least (w.r.t.  $\sqsubseteq$ )  $m \in \text{Sub}(A)$  such that  $f = m \circ e$  for some  $e : X \rightarrow M$  and we write  $\text{Im}(f) := M$ . Finally, if  $\mathbb{C}$  has coproducts, then, for a family  $(m_k)_{k \in I}$  of subobjects of  $A$ ,  $\bigcup_{k \in I} m_k$  denotes the image of  $[m_k]_{k \in I} : \coprod_{k \in I} M_k \rightarrow A$ , the latter being the unique coproduct mediator of the family  $(m_k : M_k \rightarrow A)_{k \in I}$ . Thus, in particular, for all  $p \in I$

$$m_p \sqsubseteq \bigcup_{k \in I} m_k. \tag{3}$$

---

<sup>5</sup> Whereas we obtain this result as a corollary from hereditariness, it is proved directly for underlying adhesive categories in [11].

**Definition 5 (Inverse Images and Upper Adjoints).** [7] Let  $f : A \rightarrow B$  be given in a category  $\mathbb{C}$  with pullbacks. We denote by  $f^{-1} : \text{Sub}(B) \rightarrow \text{Sub}(A)$  the inverse image function which assigns to  $m \in \text{Sub}(B)$  its pullback along  $f$ .<sup>6</sup> A monotone<sup>7</sup> function  $\mathcal{U} : \text{Sub}(A) \rightarrow \text{Sub}(B)$  is called an upper adjoint of  $f^{-1}$ , if for all  $n \in \text{Sub}(A)$  and  $m \in \text{Sub}(B)$ :

$$f^{-1}(m) \sqsubseteq n \iff m \sqsubseteq \mathcal{U}(n) \tag{4}$$

Note that  $\mathcal{U}$  is unique, if it exists [7], and that  $f^{-1}$  is monotone, since pulling back (between comma categories) is functorial and preserves monomorphisms.

4.2 ... and in  $\mathbb{CS}$

**Proposition 8.**  $\mathbb{CS}$  has images and the pullback functors preserve them.

*Proof.* Let  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{A}' = (A_i \xrightarrow{f_i} A'_i)_{0 \leq i \leq n}$  be a  $\mathbb{CS}$ -arrow. We use  $\mathbb{G}$ 's epi-mono-factorizations [6] to decompose  $f_0$  and  $(f_j)_{1 \leq j \leq n}$  in Fig. 5 accordingly. In particular  $f_0 = m_0 \circ e_0$ . Then the pullback of  $m_0$  and  $\sqsubseteq'_j$  and its unique mediator  $u$  w.r.t.  $df$  and  $e_0 \circ \sqsubseteq_j$  yields the situation in Fig. 9a, where the left upper square is a pullback by the pullback decomposition lemma.

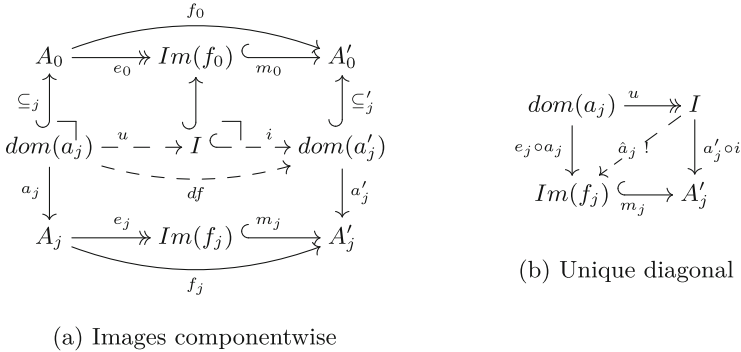


Fig. 9. Images in  $\mathbb{CS}$

In  $\mathbb{G}$  pullbacks preserve epimorphisms, i.e.  $u$  is an epimorphism and the square in Fig. 9b has a unique diagonal [6]  $\hat{a}_j : I \rightarrow \text{Im}(f_j)$ , such that everything commutes. Adding this diagonal in Fig. 9a yields  $\mathbf{Im}(\mathbf{f}) := (\text{Im}(f_0) \xrightarrow{\hat{a}_j} \text{Im}(f_j))_{1 \leq j \leq n}$  and the inclusion  $\mathbf{Im}(\mathbf{f}) \hookrightarrow \mathbf{A}'$ . Moreover,  $\mathbf{Im}(\mathbf{f})$  can be shown to

<sup>6</sup> Since we are working with equivalence classes, is easy to see that  $f^{-1}$  is independent of the choice of representative of  $m$  and independent of the choice of pullback.

<sup>7</sup> A function  $U : (X, \leq_X) \rightarrow (Y, \leq_Y)$  between two partially ordered sets is called monotone, if it preserves the order, i.e.  $\forall x, x' \in X : x \leq_X x' \Rightarrow U(x) \leq_Y U(x')$ .

be the image of  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{A}'$  in  $\mathbb{CS}$ , because it was set up by componentwise epi-mono-factorization (in  $\mathbb{G}$ ), in which the mono-part is componentwise the least subobject of the respective codomains of  $f_0$  and  $f_j$ .

Pullback functors preserve images in  $\mathbb{CS}$  because of the essential uniqueness of epi-mono-factorisations, of preservation of monomorphisms and epimorphisms [6] under pullbacks in  $\mathbb{G}$ , and componentwise pullback construction (cf. Proposition 6).  $\square$

**Proposition 9 (Upper Adjoints in  $\mathbb{CS}$ ).** *Let  $\mathbf{f} : \mathbf{A} \rightarrow \mathbf{B}$  and  $\mathbf{n} \in \text{Sub}(\mathbf{A})$ , then  $\mathcal{U}(\mathbf{n}) := \bigcup \{ \mathbf{m} \in \text{Sub}(\mathbf{B}) \mid f^{-1}(\mathbf{m}) \sqsubseteq \mathbf{n} \}$  is the upper adjoint of  $f^{-1}$ .*

*Proof.* To prove that  $\mathcal{U}$  is monotone, assume  $\mathbf{n}, \mathbf{n}' \in \text{Sub}(\mathbf{A})$  with  $\mathbf{n} \sqsubseteq \mathbf{n}'$ . Hence  $X := \{ \mathbf{m} \in \text{Sub}(\mathbf{B}) \mid f^{-1}(\mathbf{m}) \sqsubseteq \mathbf{n} \} \subseteq \{ \mathbf{m} \in \text{Sub}(\mathbf{B}) \mid f^{-1}(\mathbf{m}) \sqsubseteq \mathbf{n}' \} =: X'$  and thus there is the mediator  $\mathbf{u} : \prod_{\mathbf{m} \in X} \text{dom}(\mathbf{m}) \rightarrow \prod_{\mathbf{m} \in X'} \text{dom}(\mathbf{m})$ , such that  $\mathcal{U}(\mathbf{n}')$  becomes a factor in a decomposition of  $\prod_{\mathbf{m} \in X} \text{dom}(\mathbf{m}) \rightarrow A$ . Since  $\mathcal{U}(\mathbf{n})$  is the least of these factors, we obtain  $\mathcal{U}(\mathbf{n}) \sqsubseteq \mathcal{U}(\mathbf{n}')$ .

In equivalence (4) “ $\Rightarrow$ ” follows immediately from (3), such that it remains to prove “ $\Leftarrow$ ”. For this it is sufficient to show  $f^{-1}(\mathcal{U}(\mathbf{n})) \sqsubseteq \mathbf{n}$  for all  $\mathbf{n} \in \text{Sub}(\mathbf{A})$ , because  $f^{-1}$  is monotone. Let for this  $\mathbf{n} \in \text{Sub}(\mathbf{A})$  be arbitrary,  $\mathcal{U}(\mathbf{n}) := \bigcup_{k \in I} \mathbf{m}_k : \mathbf{J} \hookrightarrow \mathbf{B}$ , and fix some  $p \in I$ . Then there is the coproduct injection  $\mathbf{i}_p : \mathbf{M}_p \rightarrow \prod_{k \in I} \mathbf{M}_k$  and by the definition of  $\bigcup_{k \in I} \mathbf{m}_k$  in Sect. 4.1, we obtain the diagram

$$\begin{array}{ccccc}
 & & \mathbf{m}_p & & \\
 & \curvearrowright & \text{---} & \curvearrowleft & \\
 \mathbf{M}_p & \xrightarrow{\mathbf{i}_p} & \prod_{k \in I} \mathbf{M}_k & \xrightarrow{\mathbf{e}} & \mathbf{J} \hookrightarrow \mathbf{B} \\
 & & & & \cup_{k \in I} \mathbf{m}_k
 \end{array} \tag{5}$$

which is mapped by  $f^{-1}$  (interpreted as pullback functor) to the upper part of the following diagram:

$$\begin{array}{ccccccc}
 & & & f^{-1}(\mathbf{m}_p) & & & \\
 & & \curvearrowright & \text{---} & \curvearrowleft & & \\
 f^{-1}(\mathbf{M}_p) & \xrightarrow{f^{-1}(\mathbf{i}_p)} & \prod_{k \in I} f^{-1}(\mathbf{M}_k) & \xrightarrow{f^{-1}(\mathbf{e})} & f^{-1}(\mathbf{J}) & \hookrightarrow & \mathbf{A} \\
 & & & \text{---} & & & \\
 & & \mathbf{h}_p & \text{---} & \mathbf{v} & \text{---} & \mathbf{N} \\
 & & & & & & \cup_{k \in I} \mathbf{m}_k
 \end{array}$$

In this diagram,  $\mathbf{h}_p$  exists with  $\mathbf{n} \circ \mathbf{h}_p = f^{-1}(\mathbf{m}_p)$ , because  $p \in I$  and thus  $f^{-1}(\mathbf{m}_p) \sqsubseteq \mathbf{n}$  by the definition of  $\mathcal{U}$ . Note that the coproduct in (5) is mapped to  $\prod_{k \in I} f^{-1}(\mathbf{M}_k)$  and  $f^{-1}(\mathbf{i}_p)$  are the respective coproduct injections, because pullbacks preserve coproducts in  $\mathbb{G}$  and both are constructed componentwise, cf. Proposition 6. We obtain  $\mathbf{v}$  as the unique mediator out of this coproduct w.r.t. all arrows  $\{ \mathbf{h}_p \mid p \in I \}$ , i.e.  $\mathbf{v} \circ f^{-1}(\mathbf{i}_p) = \mathbf{h}_p$  and hence for all  $p \in I$ :  $\mathbf{n} \circ \mathbf{v} \circ f^{-1}(\mathbf{i}_p) = f^{-1}(\mathbf{m}_p) = f^{-1}(\bigcup_{k \in I} \mathbf{m}_k) \circ f^{-1}(\mathbf{e}) \circ f^{-1}(\mathbf{i}_p)$ , the last equality by functoriality of  $f^{-1}$ . By universality of coproducts this yields  $\mathbf{n} \circ \mathbf{v} = f^{-1}(\bigcup_{k \in I} \mathbf{m}_k) \circ f^{-1}(\mathbf{e})$ .

By Proposition 8, the latter term in this equation is the image factorisation of  $\mathbf{n} \circ \mathbf{v}$  and hence  $f^{-1}(\mathcal{U}(n)) = f^{-1}(\bigcup_{k \in I} \mathbf{m}_k) \sqsubseteq \mathbf{n}$ , the former being the least, the latter being some subobject of  $\mathbf{A}$  factoring through  $\mathbf{n} \circ \mathbf{v}$ .  $\square$

### 4.3 The Main Theorem

**Theorem 1.** *Par(CS) has all pushouts.*

*Proof.* Because CS has all pushouts by Proposition 6 and thus span-completions, this follows from Proposition 7, Corollary 1 and Proposition 9.  $\square$

Due to space limitations we can not provide an example in which the full power of SPO rewriting compared to the DPO approach can be demonstrated. Instead we provide a simple example, which reveals one additional helpful aspect of our definition of CS-morphisms. The proper construction of arbitrary pushouts in  $\text{Par}(\text{CS})$  (with non-injective rules and/or partial matches) is elaborated in [7].

Assume that a comprehensive system is in state  $\mathbf{G}$  (the bottom left system in Fig. 10) which is apparently inconsistent w.r.t **IMC** in Sect. 1, because the person in  $G_{2/3}$  possesses no phone. If it turns out that this person is the same as the one in  $G_1$ , we can restore global consistency by applying rule  $\rho : \mathbf{L} \rightarrow \mathbf{R} \in \text{Arr}(\text{Par}(\text{CS}))$ , which deletes a binary commonality (dashed) among two person records and adds a new ternary commonality which comprises these two records. The application of the rule yields a comprehensive system  $\mathbf{H}$  (bottom right), which satisfies **IMC**. Despite being relatively simple, this example demonstrates an advantage of our approach: We do not need *Negative Application Conditions* to prevent repeated application of the rule, because there is no longer a corresponding match of  $\mathbf{L}$  into  $\mathbf{H}$ . This is the case, because  $l_1$

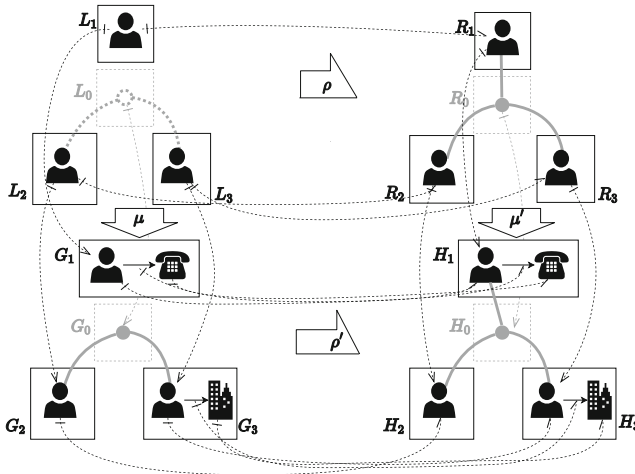


Fig. 10. SPO rule application

(the projection of  $L_0$  into  $L_1$ , cf. Definition 3) is undefined on the only element in  $L_0$ , but the respective projection  $h_1$  in  $\mathbf{H}$  is defined on the hypothetically matched element (in  $H_0$ ), i.e. the required reflection of definedness is violated, cf. Definition 4.

## 5 Related and Future Work

The best reference for *Single Pushout Rewriting* is [12], see also [2]. *Comprehensive systems* are basically a functor category invented in [20] and generalized in [11], its basic ideas originating from the theory of triple graphs [18]. Pushouts in partial map categories and especially hereditariness of colimits have been thoroughly investigated in [7, 8].

Our approach still lacks the proof that it is practically applicable, but we hope that SPO rules can serve as a basis for repair rules [15] in order to maintain consistency of informationally overlapping multimodels. We must also find an appropriate way of SPO rule typing, which can not rely on pure slice categories, because a typing morphism should not be forced to reflect definedness. And there should be a thorough characterisation of conflict-freeness in  $\mathbb{CS}$ , which is difficult, because upper adjoints in  $\mathbb{CS}$  are not constructed componentwise. The situation is as in the following quotation: “The contents of this [paper] should rather be considered a starting point . . . than the final document of this research issue”<sup>8</sup>.

**Acknowledgment.** We want to thank Tobias Heindel for pointing to the basic theoretical results needed to produce the results in this paper. Moreover, we would like to thank the referees for their constructive criticism.

## References

1. Adámek, J., Herrlich, H., Strecker, G.E.: Abstract and Concrete Categories. KATMAT: University Bremen (2004). <http://katmat.math.uni-bremen.de/acc/acc.pdf>
2. Burmeister, P., Monserrat, M., Rosselló, F., Valiente, G.: Algebraic transformation of unary partial algebras II: single-pushout approach. *Theor. Comput. Sci.* **216**(1–2), 311–362 (1999)
3. Diskin, Z., König, H., Lawford, M.: Multiple model synchronization with multiary delta lenses with amendment and K-Putput. *Formal Aspects Comput.* **31**(5), 611–640 (2019). <https://doi.org/10.1007/s00165-019-00493-0>
4. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. MTCSAES. Springer, Heidelberg (2006). <https://doi.org/10.1007/3-540-31188-2>
5. Feldmann, S., Kernschmidt, K., Wimmer, M., Vogel-Heuser, B.: Managing inter-model inconsistencies in model-based systems engineering: application in automated production systems engineering. *J. Syst. Softw.* **153**, 105–134 (2019). <https://doi.org/10.1016/j.jss.2019.03.060>. <http://www.sciencedirect.com/science/article/pii/S0164121219300639>

<sup>8</sup> This is an almost identical citation of the last statement in Michael’s PhD Thesis!



6. Goldblatt, R.: *Topoi: The Categorical Analysis of Logic*. Dover Publications, New York (1984)
7. Hayman, J., Heindel, T.: On pushouts of partial maps. In: Giese, H., König, B. (eds.) *ICGT 2014*. LNCS, vol. 8571, pp. 177–191. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-09108-2\\_12](https://doi.org/10.1007/978-3-319-09108-2_12)
8. Heindel, T.: A category theoretical approach to the concurrent semantics of rewriting: adhesive categories and related concepts. Ph.D. thesis, University of Duisburg-Essen (2009). <http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-24329/diss.pdf>
9. Heindel, T.: Hereditary pushouts reconsidered. In: Ehrig, H., Rensink, A., Rozenberg, G., Schürr, A. (eds.) *ICGT 2010*. LNCS, vol. 6372, pp. 250–265. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-15928-2\\_17](https://doi.org/10.1007/978-3-642-15928-2_17)
10. Kennaway, R.: Graph rewriting in some categories of partial morphisms. In: Ehrig, H., Kreowski, H.-J., Rozenberg, G. (eds.) *Graph Grammars 1990*. LNCS, vol. 532, pp. 490–504. Springer, Heidelberg (1991). <https://doi.org/10.1007/BFb0017408>
11. Kosiol, J., Fritsche, L., Schürr, A., Taentzer, G.: Adhesive subcategories of functor categories with instantiation to partial triple graphs. In: Guerra, E., Orejas, F. (eds.) *ICGT 2019*. LNCS, vol. 11629, pp. 38–54. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-23611-3\\_3](https://doi.org/10.1007/978-3-030-23611-3_3)
12. Löwe, M.: Extended algebraic graph transformation. Ph.D. thesis, Technical University of Berlin, Germany (1991). <http://d-nb.info/910935696>
13. Löwe, M., Tempelmeier, M.: Single-pushout rewriting of partial algebras. In: Plump, D. (ed.) *Proceedings of GCM Co-located with ICGT/STAF, L'Aquila, Italy*. CEUR Workshop Proceedings, vol. 1403, pp. 82–96 (2015). <http://ceur-ws.org/Vol-1403/paper7.pdf>
14. Löwe, M.: Algebraic approach to single-pushout graph transformation. *Theoret. Comput. Sci.* **109**(1), 181–224 (1993)
15. Rabbi, F., Lamo, Y., Yu, I.C., Kristensen, L.M.: A diagrammatic approach to model completion. In: Dingel, J., Kokaly, S., Lucio, L., Salay, R., Vangheluwe, H. (eds.) *Proceedings of the 4th Workshop on the Analysis of Model Transformations Co-located with MODELS 2015, Ottawa, Canada*. CEUR Workshop Proceedings, vol. 1500, pp. 56–65. CEUR-WS.org (2015). <http://ceur-ws.org/Vol-1500/paper7.pdf>
16. Raoult, J.: On graph rewritings. *Theor. Comput. Sci.* **32**, 1–24 (1984). [https://doi.org/10.1016/0304-3975\(84\)90021-5](https://doi.org/10.1016/0304-3975(84)90021-5)
17. Robinson, E., Rosolini, G.: Categories of partial maps. *Inf. Comput.* **79**(2), 95–130 (1988)
18. Schürr, A.: Specification of graph translators with triple graph grammars. In: Mayr, E.W., Schmidt, G., Tinhofer, G. (eds.) *WG 1994*. LNCS, vol. 903, pp. 151–163. Springer, Heidelberg (1995). [https://doi.org/10.1007/3-540-59071-4\\_45](https://doi.org/10.1007/3-540-59071-4_45)
19. Stevens, P.: Bidirectional transformations in the large. In: *MODELS 2017*, pp. 1–11 (2017)
20. Stükel, P., König, H., Lamo, Y., Rutle, A.: Multimodel correspondence through inter-model constraints. In: *Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming, Nice, France, 09–12 April 2018*, pp. 9–17 (2018). <http://doi.acm.org/10.1145/3191697.3191715>
21. Stükel, P., König, H., Lamo, Y., Rutle, A.: Towards multiple model synchronization with comprehensive systems. In: Wehrheim, H., Cabot, J. (eds.) *FASE 2020*. LNCS, vol. 12076, pp. 335–356. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45234-6\\_17](https://doi.org/10.1007/978-3-030-45234-6_17)

22. Stünkel, P., König, H., Lamo, Y., Rutle, A.: Towards multiple model synchronization with comprehensive systems: extended version. Technical report, University of Applied Sciences, FHDW Hannover (2020). <https://fhdwdev.ha.bib.de/public/papers/02020-01.pdf>
23. Trollmann, F., Albayrak, S.: Extending model synchronization results from triple graph grammars to multiple models. In: Van Van Gorp, P., Engels, G. (eds.) ICMT 2016. LNCS, vol. 9765, pp. 91–106. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-42064-6\\_7](https://doi.org/10.1007/978-3-319-42064-6_7)