

Generalised Compositionality in Graph Transformation

Amir Hossein Ghamarian and Arend Rensink

Department of Computer Science, Universiteit Twente
{a.h.ghamarian, rensink}@cs.utwente.nl

Abstract. We present a notion of composition applying both to graphs and to rules, based on graph and rule interfaces along which they are glued. The current paper generalises a previous result in two different ways. Firstly, rules do not have to form pullbacks with their interfaces; this enables graph passing between components, meaning that components may “learn” and “forget” subgraphs through communication with other components. Secondly, composition is no longer binary; instead, it can be repeated for an arbitrary number of components.

1 Introduction

We believe that, for graph transformation to become a practicable specification technique, its native strengths should be complemented with a notion of compositionality which allows the user to specify and analyse a system modularly. Failing that, graphs always have to be specified monolithically, which for large graphs quickly becomes prohibitive and causes the advantage of visualisation to be lost. Moreover, if graph transformations are used to specify the dynamic behaviour of systems, having a large monolithic graph as a state introduces the dreaded problem of state space explosion.

The issue of compositionality (or dually, modularity) has indeed been identified and addressed in a number of different approaches over the years — see Sect. 4 for an overview. Several of these, such as borrowed contexts [6], transformation units [15] and synchronised hyperedge replacement [7] have been inspired to some degree or another by notions of composition from process algebra. In this paper we continue an investigation started in [19] based on the following initial requirements:

- Composition should make it possible to construct large graphs (describing the global system in context) from smaller graphs (describing individual components).
- Composition should act as an operator over graph production systems: given a number of production systems describing individual components, the result should be a production system describing the global system. We want to introduce as little additional structure as possible.
- The behaviour of the composed system, in terms of rule applications, should likewise arise out of the composition of local system behaviour. This means that local rule applications need to be synchronised and exchange information.

In [19], we proposed to use graphs and rules with interfaces for the local systems; composition glues graphs and rules together over their interfaces using a categorical construction called *pushout*. A limitation of that setup is that components cannot exchange node or edge identities; i.e., it is impossible for one component to “publish”

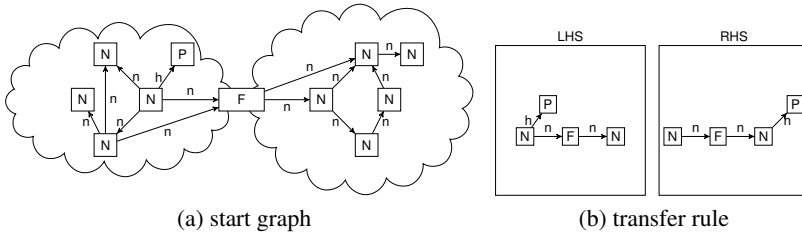


Fig. 1: Running example: Firewall between local networks

part of its internal structure and share it with others. (Instead, shared structure can only arise through simultaneous creation.)

In the current paper we lift this restriction by using a different notion of rule: rather than relying on the usual spans of morphisms we resort to *cospan*s. Though ordinary cospan rules and span rules have been shown to be equally expressive, cospan rules turn out to be advantageous in the presence of interfaces and composition. Another important difference with the usual concept of rule interface (called *kernel* in [9]) is that the relation between rule and interface is less strict (we do not insist on pullbacks).

Motivating example. To illustrate our setup, we use a running example based on two local networks which are connected via a firewall. Such a network is depicted in Fig. 1a. Each cloud represents a local network and each local network has its own network nodes, represented by \boxed{N} s. Network nodes are connected via next edges (denoted by $\boxed{N} \xrightarrow{n} \boxed{N}$) to their neighbours. Nodes can also have packets which are denoted by $(\boxed{N} \xrightarrow{n} \boxed{P})$. The firewall node (\boxed{F}) is the only interface between the local networks, through which they can communicate by sending and receiving packets. The firewall node passes safe packets through and deletes the infected packets.

Each local network may have dedicated rules to reflect its packet generation and transmission protocols. In order to avoid the state space explosion that ensues in the global network, it is desirable to specify and analyse each local network separately and obtain the global analysis by composition.

In this example, all the local structure (nodes and edges) of each network can be hidden from the other, except the firewall node which serves as the interface between the two networks. Similarly, the dynamic behaviour of the networks, given by graph rules, can also be considered local except the rule dealing with the transmission of packets from one network to another.

In the following we do not consider the local network behaviour; instead we focus on the transmission of a packet through the firewall. This behaviour is captured as a graph rule in Fig. 1b. To obtain the local effect on each network, we have to split this into two rules. An ad hoc decomposition can consist of a rule which deletes the packet node in one local network and the counterpart of this rule that creates the packet node in the other network. However, this has one major drawback: packets in the network usually have content, which is lost by the ad hoc rule decomposition. The approach of this paper enables us to pass the node itself between the decomposed systems.

Roadmap. In the next section, we give the basic definitions, especially the composition of graphs. Sect. 3 contains the main results. In Sect. 4 we review related approaches and summarise the contribution.

The basic ideas of this paper were presented for a concrete category of graphs in [8]. With respect to that paper, we have ironed out a number of technical issues and lifted the theory to the algebraic level.

2 Basic definitions

In the grand tradition of algebraic graph transformation, we develop our theory in the setting of adhesive categories. It has been shown in [16] that adhesive categories form a nice, general framework in which properties of graph transformation systems can be proved abstractly; they generalise in some part the High-Level Replacement systems studied in, e.g., [4].

Definition 1 (adhesive category). *A category \mathbf{C} is adhesive if it satisfies the following properties:*

1. \mathbf{C} has pushouts along monomorphisms (monos);
2. \mathbf{C} has pullbacks;
3. Pushouts along monos are Van Kampen squares.

For those that are not familiar with this theory, the following intuitions may be helpful:

- A mono $f: A \hookrightarrow B$ identifies a subobject of B that is isomorphic to A ;
- The pushout of $B \xleftarrow{f} A \xrightarrow{g} C$ may be thought of as the union of B and C , where the shared subset is given by A and its “embedding” in B and C ;
- The pullback of $B \xrightarrow{h} D \xleftarrow{k} C$ may be thought of as the intersection of B and C , where their “embedding” in D determines which elements they have in common.

As an example concrete category, one may think of edge-labelled directed graphs $\langle N, E, L, s, t, l \rangle$ with $s, t: E \rightarrow N$ the source and target function from the edges E to nodes N , and $l: E \rightarrow L$ a labelling function to a set of labels L . This is the context in which our running example is formulated.

In contrast with the usual setup, we take a transformation rule p not to be a span but a cospan of morphisms. In a cospan rule, creation occurs before deletion instead of the other way around. Cospan rules have been studied in [5], where the following is shown:

- A cospan rule is equivalent to the span rule arising from the pullback of the cospan;
- A span rule is equivalent to the cospan rule arising from the pushout of the span.

As a corollary, it follows that for every cospan rule there exists an equivalent cospan rule in which the morphisms are jointly epimorphic. Curiously, as we will see, cospan rules that are *not* jointly epi do play an essential role in our notion of rule composition.

Definition 2 (rule). *Let \mathbf{C} be an adhesive category.*

- A rule p consists of a cospan of monos $L \xrightarrow{l} U \xleftarrow{r} R$ where L is the left hand side, R the right hand side and U the upper object.
- The application of p on G is defined by the following diagram, where $m: L \hookrightarrow G$ is a mono (in this paper we consider monomorphic matches [11]).

$$\begin{array}{ccccc}
L & \xrightarrow{l} & U & \xleftarrow{r} & R \\
m \downarrow & PO & k \downarrow & PO & \downarrow m' \\
G & \xrightarrow{g} & K & \xleftarrow{h} & H
\end{array}$$

We write $G \xrightarrow{p,m} H$ to denote the existence of a rule application of p to G under match m , with result H .

2.1 Marked objects

We now define the general notion of a marked object, as a monomorphism from an inner object to an outer object. The inner object serves as an interface used to glue marked objects together: gluing two marked objects with the same interface comes down to taking the pushout of the corresponding span. This extends to arrows naturally.

Definition 3 (marked object and arrow). *Let \mathbf{C} be a category.*

- A marked object X is a monomorphism $e_X: \underline{X} \hookrightarrow \overline{X}$. \underline{X} is called the inner object and \overline{X} the outer object. Two marked objects X, Y are compatible if $\underline{X} = \underline{Y}$. If this is the case, we will use $X+Y$ to refer to the marked object defined by $c_Y \circ e_X: \underline{X} \hookrightarrow Z$ in the diagram

$$\begin{array}{ccc}
\overline{X} & \xrightarrow{c_Y} & Z \\
e_X \uparrow & PO & \uparrow \\
\underline{X} = \underline{Y} & \hookrightarrow & \overline{Y}
\end{array}$$

We will refer to the composed object as global and each of the original objects as local.

- Given two marked objects X, Y , a marked arrow $f: X \rightarrow Y$ is a pair of morphisms $\underline{f}: \underline{X} \rightarrow \underline{Y}$ and $\overline{f}: \overline{X} \rightarrow \overline{Y}$ such that the resulting (left hand) diagram commutes:

$$\begin{array}{ccc}
\overline{X} & \xrightarrow{\overline{f}} & \overline{Y} \\
e_X \uparrow & & \uparrow e_Y \\
\underline{X} & \xrightarrow{\underline{f}} & \underline{Y}
\end{array}
\qquad
\begin{array}{ccc}
\overline{X+Y} & \xrightarrow{h} & \overline{V+W} \\
\swarrow \wr & & \swarrow \wr \\
\overline{X} & \xrightarrow{\overline{f}} & \overline{V} \\
\downarrow PO & & \downarrow PO \\
\underline{X} = \underline{Y} & \xrightarrow{\underline{f}} & \underline{V} \\
\swarrow \wr & & \swarrow \wr \\
\underline{X} & \xrightarrow{\underline{f}} & \underline{Y} \\
\downarrow e_X & & \downarrow e_Y \\
\underline{X} & \xrightarrow{\underline{f}} & \underline{Y} \\
\downarrow e_X & & \downarrow e_Y \\
\underline{X} & \xrightarrow{\underline{f}} & \underline{Y}
\end{array}$$

Two marked arrows $f: X \rightarrow V, g: Y \rightarrow W$ are compatible if $\underline{f} = \underline{g}$. If this is the case, we will use $f + g$ to refer to the marked arrow consisting of \underline{f} and the mediating morphism h , connected by e_{X+Y} and e_{V+W} (right hand diagram above).

- A marked arrow f is called strict if \underline{f} and \overline{f} are monos and the left hand diagram above is a pullback in \mathbf{C} .

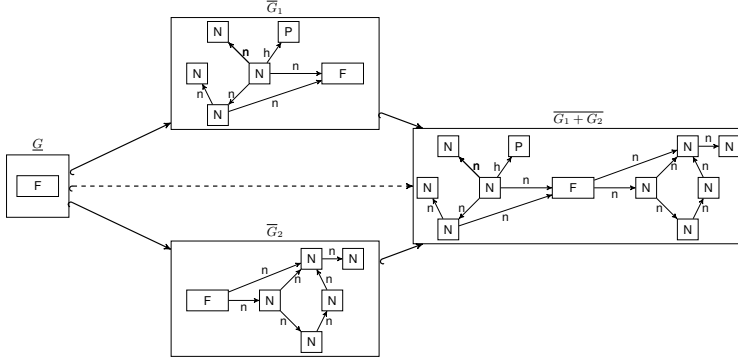


Fig. 2: Running example: composition of marked objects

Example. In our running example, the graphs associated to the local networks can be specified as compatible marked objects with the firewall node as their inner object. These marked graphs and their composition is illustrated in Fig. 2.

Given a category \mathbf{C} , we use \mathbf{C}^M to denote the category of marked \mathbf{C} -objects and -arrows. The following properties are important.

Proposition 4 (properties of marked arrows). *Let \mathbf{C} be an adhesive category.*

1. *Monos in \mathbf{C}^M correspond to pairs of monos in \mathbf{C} , stacked on top of one another. (Hence, for instance, strict arrows are monos in \mathbf{C}^M .)*
2. *Pushouts over strict arrows in \mathbf{C}^M correspond to pairs of pushouts in \mathbf{C} , stacked on top of one another.*

2.2 Marked rules

In the remainder of this paper, we will mainly deal with transformation in \mathbf{C}^M . As expected (given the above), marked rules will be cospans of marked monos. We do *not* require that the monos are strict (i.e., pullbacks). This makes the definition quite a bit more general than similar notions in [19,9], a fact which is at the core of this paper's contribution.

The intention is that a marked rule should act upon a marked object by applying the outer rule to the outer object and the inner rule to the inner object. To make this work, we have to limit matches to strict monos.

Definition 5 (marked rule and match). *Let \mathbf{C} be an adhesive category.*

- *A marked rule $p = (a, L \hookrightarrow U \hookleftarrow R)$ consists of a name a and a cospan of marked monos in \mathbf{C}^M . We write $\underline{p} = (\underline{L} \hookrightarrow \underline{U} \hookleftarrow \underline{R})$ for its inner rule and $\overline{p} = (\overline{L} \hookrightarrow \overline{U} \hookleftarrow \overline{R})$ for its outer rule.*
- *A marked match is a strict arrow in \mathbf{C}^M .*

Example. An example marked rule, transfer-1, is shown in Fig. 3 where all L_1, U_1 and R_1 are marked objects with $\overline{L}_1, \overline{U}_1, \overline{R}_1$ and $\underline{L}_1, \underline{U}_1, \underline{R}_1$ as their outer and inner graphs respectively. There are several things to be noted:

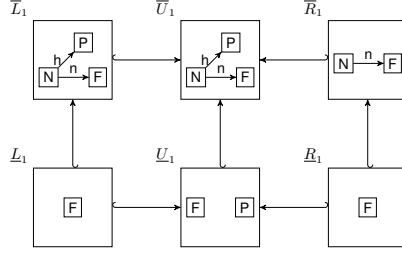


Fig. 3: Example: a marked rule transfer-1

- The inner rule $\underline{L}_1 \hookrightarrow \underline{U}_1 \leftarrow \underline{R}_1$ is not jointly epimorphic: the P-node in \underline{U}_1 is in the image of neither morphism. In fact, in this inner rule, the node is created and then immediately deleted, which does not appear to be very useful. However, it is precisely this feature that allows the node to be communicated to any other local rule with which transfer-1 is composed.
- The \mathbf{C}^M -mono $L_1 \hookrightarrow U_1$ is not strict; i.e., the corresponding square is not a pullback. Thus, deletion and creation in the inner rule do not strictly follow the outer rule. This reflects the fact that previously private parts of the graph may be (temporarily) “published” to the interface.

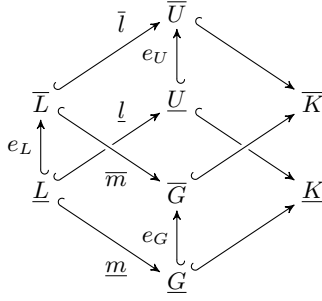
Under the assumption that \mathbf{C} is adhesive, the category of marked objects is known to be *quasi-adhesive* [14], which is a weaker notion that still retains all the nice properties of graph transformation, provided the rules are made up of strict arrows only. However, since our marked rules are not made up of strict monos, we cannot benefit from this result. In the remainder of the paper we ignore the link to quasi-adhesive categories.

The following states the first important result of this paper: the application of a marked rule to a marked object is fully characterised by the applications of the inner and outer rules. Informally speaking, the embedding morphisms of the intermediate and target (marked) objects take care of themselves.

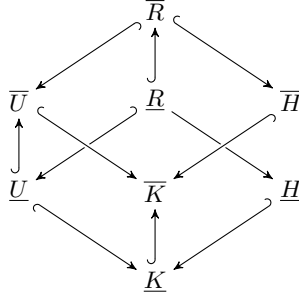
Theorem 6 (marked transformation). *Let \mathbf{C} be an adhesive category. If p is a marked rule, G a marked object and $m: L \rightarrow G$ a marked match, then $\overline{G} \xrightarrow{\overline{p}, \overline{m}} H_1$ and $\underline{G} \xrightarrow{\underline{p}, \underline{m}} H_2$ (in \mathbf{C}) if and only if $G \xrightarrow{p, m} H$ (in \mathbf{C}^M) with $\overline{H} = H_1$ and $\underline{H} = H_2$.*

This is related to the fact (Prop. 4.2) that pushout squares in \mathbf{C}^M are precisely stacks of pushout squares in \mathbf{C} for the inner and outer objects. The relevant properties are stated in the following two propositions.

Proposition 7 (pushout of marked objects). *In the following diagram, if $U \hookrightarrow L \hookrightarrow G$ is a span of marked monos, the top and bottom faces are pushouts and $m: L \hookrightarrow G$ is a marked match, then (a) there is a unique monomorphism $e_K: \underline{K} \hookrightarrow \overline{K}$ that makes the diagram commute (and which therefore makes K a marked object); and (b) the back right rectangle, $U \hookrightarrow K$, is a pullback.*



Proposition 8 (pushout complement of marked objects). *In the following diagram, if $R \hookrightarrow U$ is a marked mono and $U \hookrightarrow K$ a marked match, and the top and bottom faces are pushouts, then there is a unique monomorphism $e_H : \underline{H} \hookrightarrow \bar{H}$ that makes the diagram commute (and which therefore makes H a marked object).*



Proof (Th. 6).

If. The marked transformation $G \xrightarrow{p,m} H$ consists of a double pushout in \mathbf{C}^M , which according to Prop. 4.2 are stacked double pushouts for the inner and outer objects.

Only if. Given transformations $\underline{G} \xrightarrow{p,m} H_1$ and $\bar{G} \xrightarrow{\bar{p},\bar{m}} H_2$ with derived cospans $\underline{G} \hookrightarrow K_1 \hookleftarrow H_1$ and $\bar{G} \hookrightarrow K_2 \hookleftarrow H_2$, respectively, we know by Prop. 7 that K_1 and K_2 form a marked object K and the intermediate morphism $U \rightarrow K$ is strict; and hence by Prop. 8 that H_1 and H_2 also form a marked object H . Moreover, the resulting stacked double pushouts form a double pushout in \mathbf{C}^M according to Prop. 4.2.

Example. The application of the transfer-1 (see Fig. 3) on marked graph G_1 shown in Fig. 2 is illustrated in Fig. 4. The front faces represent the marked rule and the back face shows the marked graphs. Note that the outer rule is applied to the outer graph and the inner rule is applied to the inner graph. The network nodes that are not involved in the rule applications are omitted for simplification.

Joint epimorphism revisited. As an aside, the proposition below states that a marked cospan can be jointly epi in \mathbf{C}^M even if the inner cospan is not jointly epi in \mathbf{C} : the property only depends on the outer cospan. Moreover, joint epimorphism of rules is preserved under composition. Thus, we suspect that the results of [5] might after all still hold in this setting.

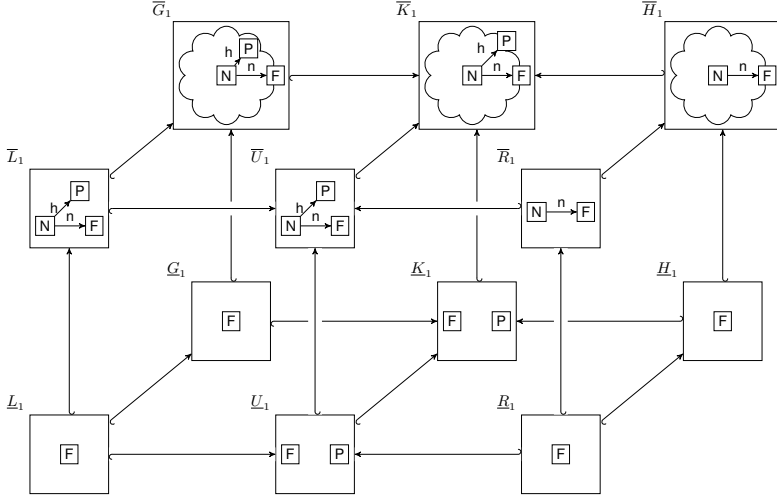


Fig. 4: Running example: application of a marked rule

Proposition 9 (joint epimorphism in \mathbf{C}^M). Let \mathbf{C} be an arbitrary category.

1. A marked cospan $X \rightarrow Z \leftarrow Y$ is jointly epimorphic in \mathbf{C}^M if and only if $\overline{X} \rightarrow \overline{Z} \leftarrow \overline{Y}$ is jointly epimorphic in \mathbf{C} .
2. If p_1, p_2 are two compatible jointly epimorphic rules in \mathbf{C}^M , then $p_1 + p_2$ is also jointly epimorphic.

3 Rule composition

We now come to the actual topic of the paper, namely the composition of rules and rule applications. First we define how rules are composed. This is entirely in line with the composition of objects and arrows in Def. 3, except that we need an additional compatibility condition.

Definition 10 (rule composition). Consider a category \mathbf{C}^M of marked objects.

- A marked rule $p = (a, L \hookrightarrow U \hookleftarrow R)$ consists of a name a and a cospan of marked monos. We write $\underline{p} = (\underline{L} \hookrightarrow \underline{U} \hookleftarrow \underline{R})$ for the inner rule and $\overline{p} = (\overline{L} \hookrightarrow \overline{U} \hookleftarrow \overline{R})$ for the outer rule.
- Two marked rules p, q are compatible if $a_p = a_q$, $\underline{p} = \underline{q}$, and \underline{L}_p resp. \underline{R}_p are the limits of the following diagrams:

$$\begin{array}{ccc}
 \overline{L}_p & \hookrightarrow & \overline{U}_p \\
 \uparrow & & \uparrow \\
 \underline{L}_p = \underline{L}_q & \hookrightarrow & \underline{U}_p = \underline{U}_q \\
 \downarrow & & \downarrow \\
 \overline{L}_q & \hookrightarrow & \overline{U}_q
 \end{array}
 \qquad
 \begin{array}{ccc}
 \overline{U}_p & \hookleftarrow & \overline{R}_p \\
 \uparrow & & \uparrow \\
 \underline{U}_p = \underline{U}_q & \hookleftarrow & \underline{R}_p = \underline{R}_q \\
 \downarrow & & \downarrow \\
 \overline{U}_q & \hookleftarrow & \overline{R}_q
 \end{array}$$

If p and q are compatible, their composition is defined by $p + q = (a, L_p + L_q \rightarrow U_p + U_q \leftarrow R_p + R_q)$, where the arrows are the composition of the left and right morphisms of p and q .

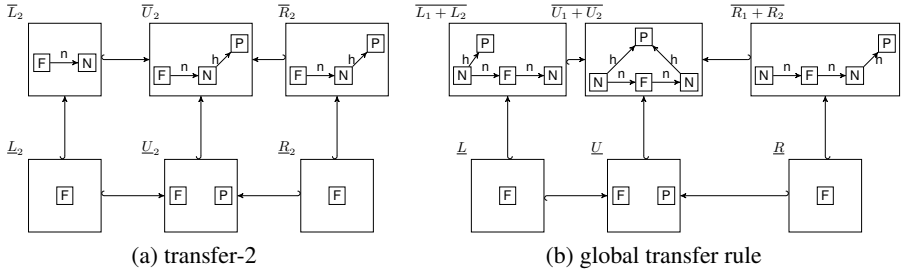


Fig. 5: Running example: composition of marked rules

It should be noted that the limit property of the inner left hand side is automatically fulfilled if one of the local left hand side morphisms is strict; and similarly for the right hand side. Thus, in a sense, this is the price we pay for relaxing our rules to allow non-strict morphisms.

Under this notion of compatibility, rule composition is well-defined, i.e., always yields a rule. In particular, we have to establish that the composed rule morphisms are monic.

Proposition 11 (marked rule). *The composition of two compatible marked rules is a marked rule.*

Example. The marked rule, transfer-2, depicted in Fig. 5a is compatible with transfer-1 (see Fig. 3): they share the same inner rule. For transfer-2 it is $R_2 \hookrightarrow U_2$ that is not a pullback: after the P-node has been received, it is removed again from the interface and becomes local to this component. The composition of transfer-1 and transfer-2 is illustrated in Fig. 5b. Note that the outer rule of Fig. 5b is the same as our original transfer rule given in Fig. 1b.

Finally, we extend composition to rule applications. When compatible rules are applied to compatible graphs under compatible matches, we also call the entire transformations compatible.

Definition 12 (transformation compatibility). *Let \mathbf{C} be an adhesive category. We call two marked transformations $G_1 \xrightarrow{p_1, m_1} H_1$ and $G_2 \xrightarrow{p_2, m_2} H_2$ (in \mathbf{C}^M) compatible if (1) p_1 and p_2 are compatible rules, (2) G_1 and G_2 are compatible objects and (3) $\underline{m}_1 = \underline{m}_2$.*

3.1 Soundness

The soundness of composition essentially states that, given two compatible transformations, the following recipes give rise to the same result:

- Compose the local rules into a single global rule and the local objects to a global object, then apply the rule (under the composed match);
- Apply the local rules to the local objects, then compose the target objects.

This can be succinctly summarised as “local behaviour generates global behaviour”.

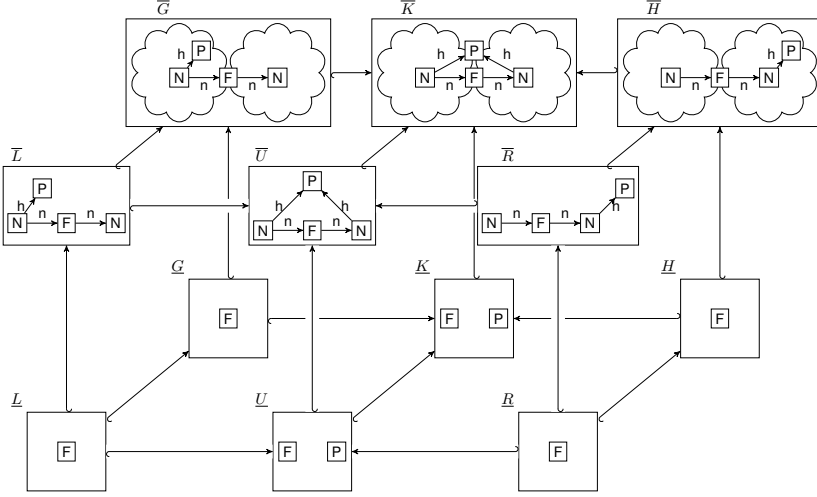


Fig. 6: Running example: composition of the transfer rule

Theorem 13 (soundness). Let \mathbf{C} be an adhesive category. If $G_i \xrightarrow{p_i, m_i} H_i$ (in \mathbf{C}^M) for $i = 1, 2$ are compatible marked transformations, then $G_1 + G_2 \xrightarrow{p_1 + p_2, m_1 + m_2} H_1 + H_2$ (in \mathbf{C}^M).

The proof is essentially due to the fact that all compositions are by pushout, and pushouts commute. Moreover, due to Th. 6 we can separately concentrate on the inner and outer part of the global transformation; and since the inner part is identical for the local and global rules, there is nothing to be shown.

Example. In Fig. 4 we have seen that K_1 and H_1 obtained by applying the transfer-1 rule (Fig. 3) to G_1 (Fig. 2). Similarly, by applying transfer-2 (Fig. 5a) to G_2 (Fig. 2) we can obtain K_2 and H_2 . It is not difficult to verify that K and H in Fig. 6 obtained by application of the composed rule (Fig. 5b) to G (Fig. 2) are in fact equivalent to $K_1 + K_2$ and $H_1 + H_2$ respectively. Moreover, the application of transfer-1 and transfer-2 allows node P to be transferred from G_1 to H_2 , while its content is preserved.

3.2 Completeness

Completeness is the dual of soundness, and can be summarised as “all global behaviour arises from local behaviour”. The proof entails showing that every global graph transformation can be decomposed into local transformations, for an arbitrary decomposition of the host graph into local graphs.

In fact, there are generally many possible decompositions of the rules, and the proof of the completeness theorem is mainly a matter of picking an appropriate candidate. For instance, if a global rule creates a graph element, then at least one of the local rules should do so. Therefore, to show the completeness property, we define an specific decomposition of a global marked rule, which guarantees the applicability of the decomposed rules.

Definition 14 (strict rule decomposition). Let p, p_1 , and p_2 be marked rules. We call p_1 and p_2 a strict decomposition of p if $p = p_1 + p_2$ and the following diagrams are pullbacks.

$$\begin{array}{ccccc} \bar{R}_1 & \hookrightarrow & \bar{R} & \longleftarrow & \bar{R}_2 \\ \downarrow & & \downarrow & & \downarrow \\ \bar{U}_1 & \hookrightarrow & \bar{U} & \longleftarrow & \bar{U}_2 \end{array} \quad \begin{array}{c} PB \\ PB \end{array}$$

The intention of insisting on the extra pullback condition, $U \leftarrow R$, in the definition of strict decomposition is to state the conditions under which the decomposed rules do not delete elements which are preserved by the global rule. In other words, if an element is preserved by the application of a global rule then it is also preserved by the application of both strictly decomposed ones. Note that this decomposition does not capture all possible rule decompositions. For instance, the decomposition of the transfer rule in our example is not strict: node P is deleted in transfer-1 while it is preserved by the global rule. For completeness, however, it turns out to be sufficient to use strict decompositions only.

This is convenient because strict decomposition guarantees the applicability of the decomposed rules whenever the original global marked rule is applicable. To show this property, first we prove the following lemma.

Lemma 15. Let \mathbf{C} be an adhesive category, p a marked rule where $U \leftarrow R$ is strict, G a marked object and m a marked matching (in \mathbf{C}^M). If there exists a transformation $\bar{G} \xrightarrow{\bar{p}, \bar{m}} \bar{H}$ (in \mathbf{C}), then there is a marked transformation $G \xrightarrow{p, m} H$ (in \mathbf{C}^M).

Now we show the conditions where a marked transformation can be decomposed to two compatible marked transformations.

Lemma 16. Let \mathbf{C} be an adhesive category, $G \xrightarrow{p, m} H$ (in \mathbf{C}^M) a marked transformation, G_1 and G_2 a decomposition of G , and p_1 and p_2 a strict rule decomposition of p , such that \bar{L}_1 and \bar{L}_2 are the pullbacks of the following diagrams.

$$\begin{array}{ccccc} \bar{L}_1 & \hookrightarrow & \bar{L} & \longleftarrow & \bar{L}_2 \\ \downarrow & & \downarrow & & \downarrow \\ \bar{G}_1 & \hookrightarrow & \bar{G} & \longleftarrow & \bar{G}_2 \end{array} \quad \begin{array}{c} PB \\ PB \end{array}$$

Then there are two compatible marked transformations $G_i \xrightarrow{p_i, m_i} H_i$ for $i = 1, 2$ such that $H = H_1 + H_2$.

In fact Lemma 16 states that a marked transformation $G \xrightarrow{p, m} H$ can be decomposed to two compatible transformations for any strict decomposition of p as long as they have valid matches. Now we have come to another main contribution of the paper. The completeness theorem states that given a global transformation, and a decomposition of its start object, we can always decompose the transformation to two compatible marked transformations according to the given start object decomposition. To prove the completeness theorem, we only need to show the existence of two such marked transformations.

Theorem 17 (completeness). *Let \mathbf{C} be an adhesive category, $G = G_1 + G_2$ be a decomposition of a marked object G . A transformation $\overline{G} \xrightarrow{p,m} H$ (in \mathbf{C}) can be decomposed to two compatible marked transformations $G_1 \xrightarrow{p_1,m_1} H_1$ and $G_2 \xrightarrow{p_2,m_2} H_2$ (in \mathbf{C}^M) such that $\overline{p_1 + p_2} = p$ and $H = \overline{H_1 + H_2}$.*

4 Conclusion

We have defined a notion of composition for graphs and graph transformation rules, in the setting of adhesive categories, which allows passing subgraphs between components. This was done by equipping every graph and graph transformation rule with an interface, which declare the parts that are exposed to the environment. Graphs and rules can be composed when they have compatible interfaces. The contributions involved are:

- Rule composition both preserves transformations from the local to the global level (soundness, see Th. 13) and reflects them from the global to the local level (completeness, see Th. 17). There are no further restrictions on the rules: the results are completely general.
- Our rule interfaces, in contrast to what we have seen elsewhere, do not have to form pullbacks with the main rule. This means that deletion and creation in the interface does not strictly follow that in the main rule. This is essential to the framework, since it enables rules to publish part of their inner structure to the outside world (via the interface).
- With respect to [19] we have changed from span rules to cospan rules, which makes the framework quite a bit more expressive; in particular, the notion of graph passing answers one of the items identified as future work in that paper.
- With respect to [8], where the concept of graph passing was presented in a concrete category of graphs, we have lifted the framework to the prevailing algebraic setting.

4.1 Related work

The concepts of graph and rule composition, with the appropriate notions of soundness and completeness, were introduced in [19] and later generalised in [12]. With respect to those papers, the variation studied here offers a more powerful notion of composition, in which nodes and edges can be deleted in one component and simultaneously created in the other.

In addition, there are a number of other approaches to introduce aspects of compositionality into graph transformation.

Synchronised Hyperedge Replacement. This is a paradigm in which graph transformation rules (more specifically, hyperedge replacement rules) can be synchronised based on the adjacency of their occurrences within a graph; see [13,7]. The synchronised rules are not themselves understood as graph transformation rules, and consequently the work does not address the type of compositionality issues that we have studied here. Still, it is interesting to see whether SHR synchronisation can be understood as a special type of composition in our sense.

History-Dependent Automata. This is a behavioural model in which states are enriched with a set of *names* (see [18] for an overview). Transitions expose names to the environment, and can also record the deletion, creation and permutation of names. HD-automata can be composed while synchronising their transitions: this provides a model for name passing. Transition systems induced by graph transformation rules can be understood as a variant of HD-automata where the states are enriched with graphs rather than just sets, and the information on the transitions is extended accordingly.

Rule amalgamation and distributed graph transformation. Studied in [3] and later, more extensively, in [20], the principle of rule amalgamation provides a general mechanism for rule (de)composition. This is a sub-problem of the one we have addressed here, as we study composition of the graphs as well as the rules. Our notion of rule composition is actually a generalisation of rule amalgamation, as local rules do not have to synchronise on deletions and creations.

Borrowed contexts. Like our paper, the work on borrowed contexts [6,1] uses a setting where only part of a graph is available, and studies the application of rules to such subgraphs in a way that is compatible with the original, reductive semantics. In contrast to our approach, however, they do not decompose rules: instead, when a rule is applied to a graph in which some of the required structure (“context”) for the match is missing, this is imported (“borrowed”) as part of the transformation. As a result, in this paradigm the subgraphs grow while being transformed, incorporating ever more context information. This is quite different from the basic intuitions behind our approach.

Summarising, where only rules are (de)composed in rule amalgamation, and only graphs in borrowed contexts, in our approach both rules and graphs are subject to (de)composition.

Compositional model transformation. [2] studies a notion of compositionality in model transformation. Though on the face of it this sounds similar, in fact they study a different question altogether, namely whether a transformation affects the *semantics* of a model (given as a separate mapping to a semantic domain) in a predictable (compositional) manner. This is in sharp contrast with our work, which rather addresses the compositionality of the graph transformation framework itself.

Graph Transformation Units. The graph transformation units exemplified in [15], also provide a notion of composition. However, this work takes the form of an explicit structuring mechanism of local graph transformation systems, called Units. The question of equivalence of a monolithic graph transformation system and a composition of local units is not addressed in this approach.

4.2 Future work

Though with this paper we have addressed a major outstanding question of [19], there is still a lot of work to be done before the compositional framework can be used in practice.

For instance, negative application conditions (NACs) as introduced in [10] have shown to be very useful in practice. It will be interesting to extend our notion of compositionality to rules with NACs, in particular with respect to the soundness and completeness properties.

Another important problem is finding an automatic mechanism for splitting the start graph, and decomposing the rule system such that both the number of states and the number of required rules for the local systems stays minimal. This was also discussed in some detail in [8], where we proposed to use *partial graphs* for this purpose; however, so far these lack a good definition on the algebraic level.

Finally, composition as introduced here is only part of the story. Again inspired by process algebra, in particular the *hiding* operator, it makes sense to think of ways in which to restrict the interface of a marked rule or graph, thus making part of the previously published interface structure private. Also, if we want to compose rules whose interfaces do not quite match, one may think of a *partial composition* operator, using ideas of borrowed contexts [6,1] (see above).

4.3 Acknowledgement

This paper could not have been written without the invaluable help of Barbara König and members of her group, in particular Matthias Hülsbusch, who provided essential hints in the proof of Th. 6.

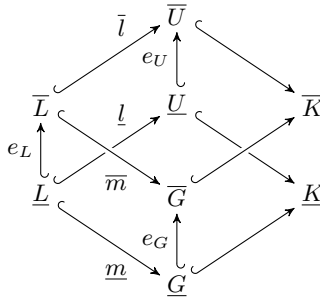
References

1. Baldan, P., Ehrig, H., König, B.: Composition and decomposition of DPO transformations with borrowed context. In: Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G. (eds.) Third International Conference on Graph Transformations, (ICGT). LNCS, vol. 4178, pp. 153–167. Springer (2006)
2. Bisztray, D., Heckel, R., Ehrig, H.: Compositionality of model transformations. In: Aldini, A., ter Beek, M., Gadducci, F. (eds.) 3rd International Workshop on Views On Designing Complex Architectures (VODCA). ENTCS, vol. 236, pp. 5–19 (2009)
3. Boehm, P., Fonio, H.R., Habel, A.: Amalgamation of graph transformations: A synchronization mechanism. *J. Comput. Syst. Sci.* 34(2/3), 377–408 (1987)
4. Ehrig, H., Habel, A., Kreowski, H.J., Parisi-Presicce, F.: From graph grammars to high level replacement systems. In: Ehrig, H., Kreowski, H.J., Rozenberg, G. (eds.) *Graph-Grammars and Their Application to Computer Science*. LNCS, vol. 532, pp. 269–291. Springer (1991)
5. Ehrig, H., Hermann, F., Prange, U.: Cospan dpo approach: An alternative for dpo graph transformations. *Bulletin of the EATCS* 98, 139–149 (2009)
6. Ehrig, H., König, B.: Deriving bisimulation congruences in the DPO approach to graph rewriting with borrowed contexts. *Mathematical Structures in Computer Science* 16(6), 1133–1163 (2006)
7. Ferrari, G.L., Hirsch, D., Lanese, I., Montanari, U., Tuosto, E.: Synchronised hyperedge replacement as a model for service oriented computing. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.P. (eds.) *Formal Methods for Components and Objects (FMCO)*. LNCS, vol. 4111, pp. 22–43. Springer (2006)
8. Ghamarian, A., Rensink, A.: Graph passing in graph transformation. In: Fish, A., Lambers, L. (eds.) *Graph Transformation and Visual Modelling Techniques*. ECEASST (2012), to be published
9. Golas, U., Ehrig, H., Habel, A.: Multi-amalgamation in adhesive categories. In: Ehrig, H., Rensink, A., Rozenberg, G., Schürr, A. (eds.) *International Conference on Graph Transformations (ICGT)*. LNCS, vol. 6372, pp. 346–361. Springer (2010)

10. Habel, A., Heckel, R., Taentzer, G.: Graph grammars with negative application conditions. *Fundam. Inform.* 26(3/4), 287–313 (1996)
11. Habel, A., Müller, J., Plump, D.: Double-pushout graph transformation revisited. *Mathematical Structures in Computer Science* 11(5), 637–688 (2001)
12. Heindel, T.: Structural decomposition of reactions of graph-like objects. In: Aceto, L., Sobocinski, P. (eds.) *Structural Operational Semantics (SOS)*. EPTCS, vol. 32, pp. 26–41 (2010)
13. Hirsch, D., Montanari, U.: Synchronized hyperedge replacement with name mobility. In: Larsen, K.G., Nielsen, M. (eds.) *Concurrency Theory (CONCUR)*. LNCS, vol. 2154, pp. 121–136. Springer (2001)
14. Johnstone, P.T., Lack, S., Sobocinski, P.: Quasitoposes, quasiadhesive categories and Artin glueing. In: Mossakowski, T., Montanari, U., Haveraaen, M. (eds.) *Algebra and Coalgebra in Computer Science (CALCO)*. LNCS, vol. 4624, pp. 312–326. Springer (2007)
15. Kreowski, H.J., Kuske, S., Rozenberg, G.: Graph transformation units - an overview. In: Degano, P., De Nicola, R., Meseguer, J. (eds.) *Concurrency, Graphs and Models*. LNCS, vol. 5065, pp. 57–75. Springer (2008)
16. Lack, S., Sobocinski, P.: Adhesive categories. In: Walukiewicz, I. (ed.) *Foundations of Software Science and Computation Structures, (FoSSaCS)*. LNCS, vol. 2987, pp. 273–288. Springer (2004)
17. Lack, S., Sobocinski, P.: Adhesive and quasiadhesive categories. *ITA* 39(3), 511–545 (2005)
18. Montanari, U., Pistore, M.: History-dependent automata: An introduction. In: Bernardo, M., Bogliolo, A. (eds.) *Formal Methods for Mobile Computing*. LNCS, vol. 3465, pp. 1–28. Springer (2005)
19. Rensink, A.: A first study of compositionality in graph transformation. Tech. Rep. TR-CTIT-10-08, Centre for Telematics and Information Technology, University of Twente (2010)
20. Taentzer, G.: Parallel high-level replacement systems. *TCS* 186(1-2), 43–81 (1997)

A Proofs of the remaining results

Proposition 7 *In the following diagram, if $U \leftarrow L \hookrightarrow G$ is a span of marked monos, the top and bottom faces are pushouts and $m: L \hookrightarrow G$ is a marked match, then (a) there is a unique monomorphism $e_K: \underline{K} \hookrightarrow \overline{K}$ that makes the diagram commute (and which therefore makes K a marked object); and (b) the back right rectangle, $U \hookrightarrow K$, is a pullback.*



Proof. First of all, the existence of a morphism $e_K: \underline{K} \rightarrow \overline{K}$ is guaranteed by the fact that the bottom square is a pushout. At this point, however, we do not know e_K to be monic. We use the following two auxiliary diagrams.

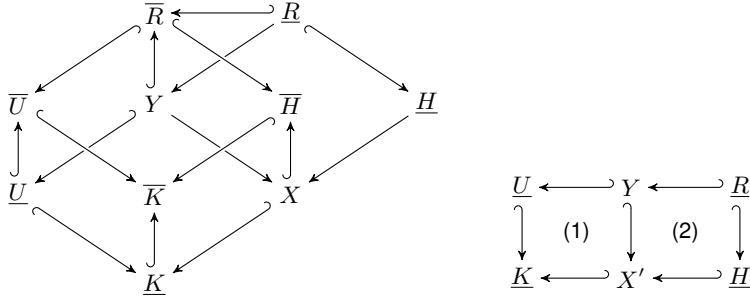
$$\begin{array}{ccc}
\underline{L} \hookrightarrow \overline{L} \hookrightarrow \overline{U} & & \underline{L} \hookrightarrow \underline{U} \hookrightarrow \overline{U} \\
\downarrow (1) & \downarrow (2) & \downarrow (3) \\
\underline{G} \hookrightarrow \overline{G} \hookrightarrow \overline{K} & & \underline{G} \hookrightarrow \underline{K} \hookrightarrow \overline{K} \\
& & \downarrow (4)
\end{array}$$

We know that the front left and top face are a pullback and pushout respectively (squares 1 and 2). Therefore, the outer square, (1)+(2), is a pullback. Moreover, the bottom face (3) is also a pushout and since all vertical morphisms are monos, according to special PO-PB property (see [17, Lemma. 8.6]) square (4) is a pullback. This means e_K is a gap morphism (see [16, Th. 17]: (3)+(4) is a pullback, (3) is pushout and all morphisms are monos), and therefore monic.

Proposition 8 *In the following diagram, if $R \hookrightarrow U$ is a marked mono and $U \hookrightarrow K$ a marked match, and the top and bottom faces are pushouts, then there is a unique monomorphism $e_H: \underline{H} \hookrightarrow \overline{H}$ that makes the diagram commute (and which therefore makes H a marked object).*

$$\begin{array}{ccccc}
& & \overline{R} & & \\
& \swarrow & \uparrow & \searrow & \\
\overline{U} & & \underline{R} & & \overline{H} \\
\uparrow & \swarrow & \downarrow & \searrow & \\
\underline{U} & & \underline{K} & & \underline{H} \\
& \searrow & \uparrow & \swarrow & \\
& & \underline{K} & &
\end{array}$$

Proof. First we make a VK cube by taking the pullbacks of $\underline{U} \hookrightarrow \overline{U} \hookrightarrow \overline{R}$ (calling the object Y) and $\underline{K} \hookrightarrow \overline{K} \hookrightarrow \overline{H}$ (calling the object X). The result is illustrated in the diagram below. The morphism between $Y \rightarrow X$ is the mediating morphism. The bottom face automatically becomes a pushout owing to the VK square property. Subsequently, there exists a mediating morphism $\underline{R} \rightarrow Y$ such that all subdiagrams commute. Now, if we alternatively construct the pushout of $Y \leftarrow \underline{R} \hookrightarrow \underline{H}$ and call the object X' , then there will be a morphism $X' \rightarrow K$. By pushout decomposition, since (1)+(2) and (2) are both pushouts, we know that (1) is a pushout. Therefore, by uniqueness of pushout complement, w.l.o.g, we can assume $X = X'$. This proves the existence of a commuting morphism $e_H: \underline{H} \rightarrow \overline{H}$. As $\underline{H} \hookrightarrow \underline{K} \hookrightarrow \overline{K}$ is a mono, by mono decomposition we can also deduce that e_H is monic.

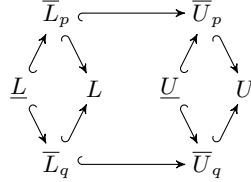


Proposition 11 *The composition of two compatible marked rules is a marked rule.*

Proof. Let p and q be two compatible marked rules, and let $p + q = (a, L \rightarrow U \leftarrow R)$. In the remainder we denote $\underline{L} = \underline{L}_p = \underline{L}_q$ and likewise for U and R .

We need to show that $L \hookrightarrow U \leftarrow R$ is a cospan of monos. We show this for $L \hookrightarrow U$; the proof of $U \leftarrow R$ is symmetrical.

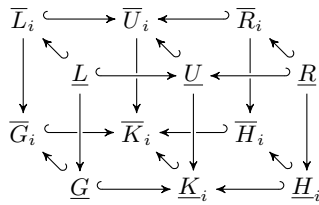
We know that \underline{L} is the limit of $\bar{L}_p \hookrightarrow \bar{U}_p \hookrightarrow \underline{U} \hookrightarrow \bar{U}_q \hookrightarrow \bar{L}_q$ by the compatibility p and q , and U is the the pushout of $\bar{U}_p \hookrightarrow \underline{U} \hookrightarrow \bar{U}_q$. In adhesive categories we know that pushouts along monomorphisms are also pullbacks. So, if we construct the pullback of the outer cospan ($\bar{L}_p \hookrightarrow \bar{U}_p \hookrightarrow U \leftarrow \bar{U}_q \hookrightarrow \bar{L}_q$) in the diagram below and call the object L' , it is not hard to see that L' also becomes the limit of the original diagram.



By the uniqueness property of limit we can assume that L' and \underline{L} are the same object. It follows that the mediating morphism between L and U becomes the gap morphism and is therefore monic.

Theorem 13 *Let \mathbf{C} be an adhesive category. If $G_i \xrightarrow{p_i, m_i} H_i$ (in \mathbf{C}^M) for $i = 1, 2$ are compatible marked transformations, then $G_1 + G_2 \xrightarrow{p_1 + p_2, m_1 + m_2} H_1 + H_2$ (in \mathbf{C}^M).*

Proof. As before, de denote $\underline{L} = \underline{L}_1 = \underline{L}_2$, and similar for U , R , G , K and H . First consider the two original marked transformations.

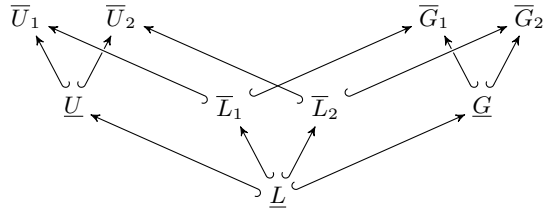


Remember that all front and back faces are pushouts. It follows that both \underline{K}_1 and \underline{K}_2 [both \underline{H}_1 and \underline{H}_2] are the pushout [pushout complement] of the same diagrams; therefore, isomorphic and w.l.o.g we may assume them to be the same objects, which implies that K_1 and K_2 [H_1 and H_2] are compatible. In the following, we denote $p = p_1 + p_2 = (L \hookrightarrow U \hookrightarrow R)$, $G = G_1 + G_2$, $K = K_1 + K_2$ and $H = H_1 + H_2$.

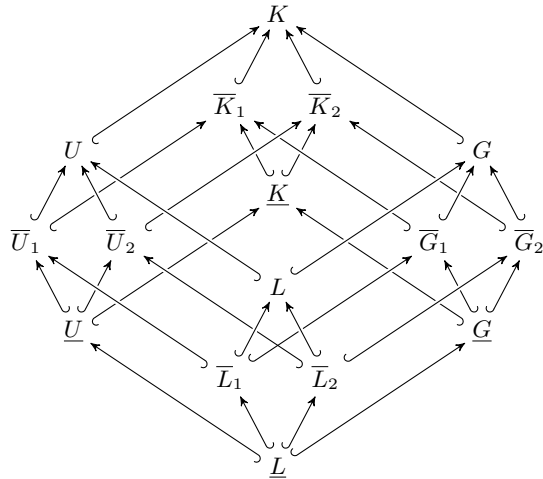
We need to show that both squares in the following diagram are pushouts:

$$\begin{array}{ccccc}
 L & \hookrightarrow & U & \hookleftarrow & R \\
 \downarrow & & \downarrow & & \downarrow \\
 & \text{PO} & & \text{PO} & \\
 G & \hookrightarrow & K & \hookleftarrow & H
 \end{array} \tag{1}$$

For the left square, this follows from the fact that K is the colimit of the following diagram:



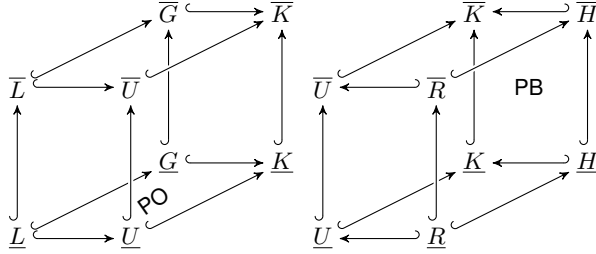
We may first construct the pushouts of $\underline{G} \hookrightarrow \underline{L} \hookrightarrow \underline{U}$, $\overline{G}_1 \hookrightarrow \overline{L}_1 \hookrightarrow \overline{U}_1$, and $\overline{G}_2 \hookrightarrow \overline{L}_2 \hookrightarrow \overline{U}_2$, which are \underline{K} , \overline{K}_1 , and \overline{K}_2 , respectively, and then recall that K is the pushout of $\overline{K}_1 \hookrightarrow \underline{K} \hookrightarrow \overline{K}_2$. But the colimit of the diagram can alternatively be constructed by taking the push out of $L \hookrightarrow I \hookrightarrow K$, which in turn are the pushout objects of, $\overline{L}_1 \hookrightarrow \underline{L} \hookrightarrow \overline{L}_2$, $\overline{U}_1 \hookrightarrow \underline{U} \hookrightarrow \overline{U}_2$ and $\overline{G}_1 \hookrightarrow \underline{G} \hookrightarrow \overline{G}_2$. Due to the universality of the colimit, the results of these two constructions coincide. The situation is again visualized in the following diagram.



For the right hand square of (1), exactly the same argument can be used, by the appropriate substitutions of L by R and G by H .

Lemma 15 *Let \mathbf{C} be an adhesive category, p a marked rule where $U \hookrightarrow R$ is strict, G a marked object and m a marked matching (in \mathbf{C}^M). If there exists a transformation $\overline{G} \xrightarrow{p,m} \overline{H}$ (in \mathbf{C}), then there is a marked transformation $G \xrightarrow{p,m} H$ (in \mathbf{C}^M).*

Proof. We construct the pushout of $\underline{G} \hookrightarrow \underline{L} \hookrightarrow \underline{U}$ (calling the object \underline{K}). Now, according to Prop. 7 \underline{K} is a marked object, i.e., there is a mono between \underline{K} to \overline{K} , and the right face is a pullback.



Now, we construct the pullback of $\underline{K} \hookrightarrow \overline{K} \hookrightarrow \overline{H}$ (calling the object \underline{H}). There is a mediating morphism between \underline{R} and \underline{H} such that all subdiagrams commute. We know that the front face and the left face are pullbacks, so is their composition. Also the back face is a pullback and by pullback decomposition the right face becomes a pullback. Consequently, all side faces are pullback and the top face is a pushout, therefore, the bottom face is a pushout. This shows that \underline{H} is the target object of the transformation of \underline{G} by applying the rule $\underline{L} \hookrightarrow \underline{U} \hookrightarrow \underline{R}$. Now, by Th. 6 we can conclude that $G \xrightarrow{p,m} H$.

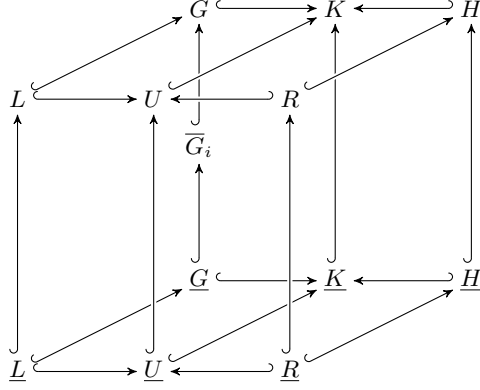
Lemma 16 *Let \mathbf{C} be an adhesive category, $G \xrightarrow{p,m} H$ (in \mathbf{C}^M) a marked transformation, G_1 and G_2 a decomposition of G , and p_1 and p_2 a strict rule decomposition of p , such that \overline{L}_1 and \overline{L}_2 are the pullbacks of the following diagrams.*

$$\begin{array}{ccccc} \overline{L}_1 & \hookrightarrow & \overline{L} & \overline{L} & \longleftarrow & \overline{L}_2 \\ \downarrow & & \downarrow & \downarrow & & \downarrow \\ \overline{G}_1 & \hookrightarrow & \overline{G} & \overline{G} & \longleftarrow & \overline{G}_2 \end{array}$$

Then there are two compatible marked transformations $G_i \xrightarrow{p_i, m_i} H_i$ for $i = 1, 2$ such that $H = H_1 + H_2$.

Proof. We are given a global marked transformation $G \xrightarrow{p,m} H$ which consist of two stacked pushouts. We need to show the existence of two compatible marked transformations. Since p_1 and p_2 are compatible and G_1 and G_2 are a decomposition of G , if there exist two marked transformations $G_1 \xrightarrow{p_1, m_1} H_1$ and $G_2 \xrightarrow{p_2, m_2} H_2$ such that they share the same inner object transformation, they are also compatible.

The following diagram illustrates the global transformation together with the decomposition of G . Because of symmetry only one object decomposition, G_i , is depicted. We now show the existence of local marked transformations.

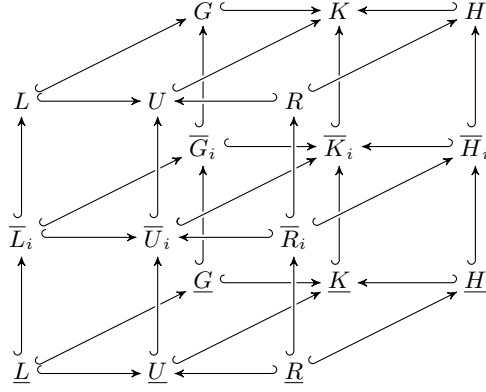


Suppose p' is a marked rule with \bar{p} and \bar{p}_i as its outer and inner rules respectively and similarly let G' be marked object with \bar{G} and \bar{G}_i as its outer and inner objects. Now because we have $G \xrightarrow{p, m} H$ we know that there exists $\bar{G} \xrightarrow{\bar{p}, \bar{m}} \bar{H}$. By Lemma 15 we can conclude that there is a marked transformation $G' \xrightarrow{p', m'} H'$, and by Th. 6 we have $\bar{G}_i \xrightarrow{\bar{p}_i, \bar{m}_i} \bar{H}_i$.

We also need to show that \bar{K}_i and \bar{H}_i are also marked objects with \underline{K} and \underline{H} as their inner objects respectively. Considering the following diagram, we know that (1)+(2) is a pullback because of the applicability condition of the global rule, and (2) is also a pullback by construction, therefore (1) becomes a pullback by the pullback decomposition property.

$$\begin{array}{ccc}
 \underline{L} & \hookrightarrow & \underline{G} \\
 \downarrow & (1) & \downarrow \\
 \underline{L}_i & \hookrightarrow & \underline{G}_i \\
 \downarrow & (2) & \downarrow \\
 L & \hookrightarrow & G
 \end{array}$$

So G_i, L_i and U_i can be considered marked objects with $\underline{G}, \underline{L}$ and \underline{U} as their inner objects. Now, by Th. 6 Now, we can conclude that H_i is the target marked object obtained by applying the marked rule p_i to G_i . The complete transformation decomposition is illustrated in the following diagram. We showed that $G_1 \xrightarrow{p_1, m_1} H_1$ and $G_2 \xrightarrow{p_2, m_2} H_2$, are compatible transformations, by Th. 13 we can conclude that $H = H_1 + H_2$.



Theorem 17 Let \mathbf{C} be an adhesive category, $G = G_1 + G_2$ be a decomposition of a marked object G . A transformation $\overline{G} \xrightarrow{p, m} H$ (in \mathbf{C}) can be decomposed to two compatible marked transformations $G_1 \xrightarrow{p_1, m_1} H_1$ and $G_2 \xrightarrow{p_2, m_2} H_2$ (in \mathbf{C}^M) such that $\overline{p_1 + p_2} = p$ and $H = \overline{H_1 + H_2}$.

Proof. Let p be $L \hookrightarrow U \leftarrow R$. We define two compatible marked rules p_1 and p_2 and their composition $p' = p_1 + p_2$ such that the outer rule of their composition, $\overline{p'}$, is equal to p . First, we define their left hand sides by constructing the pullbacks of $L \hookrightarrow \overline{G} \leftarrow \overline{G}_1$ and $L \hookrightarrow \overline{G} \leftarrow \overline{G}_2$ and calling the pullback objects L_1 and L_2 respectively. We also construct L_{12} by taking the pullback of $L_1 \hookrightarrow L \leftarrow L_2$. Now, we define p_i with $\overline{p}_i = L_i \hookrightarrow U \leftarrow R$ and $\underline{p}_i = L_{12} \hookrightarrow U \leftarrow R$ for $i = 1, 2$. By construction we know that p_1 and p_2 are compatible and also the outer rule of their composition, $\overline{p'}$, is equal to p . Moreover, p_1 and p_2 are in fact a strict decomposition of p' , since $U \leftarrow R$ are pullbacks in both rules. Now, from Lemma 15 it follows that there is a marked transformation $G_1 + G_2 \xrightarrow{p', m} H'$, where $\overline{H'} = H$. Subsequently, due to Lemma 16 we know that there are two marked transformations $G_1 \xrightarrow{p_1, m_1} H_1$ and $G_2 \xrightarrow{p_2, m_2} H_2$ such that $H' = H_1 + H_2$. By Th. 6 we can deduce that $H = \overline{H_1 + H_2}$.