REPORTS
IN
INFORMATICS

ISSN 0333-3590

The Diagram Predicate Framework
in View of Adhesive Categories

Uwe Wolter, Department of Informatics,
University of Bergen, Norway
Florian Mantz, Bergen University College,
Norway

REPORT NO 405 **August 2013**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/pdf/2013-405.pdf

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at
http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Contents

Abstract

Graph transformation is the rule-based manipulation of graphs, a concept that becomes increasingly important in computer science. The theory of Adhesive High-Level Replacement (HLR) systems generalizes the classical theory of algebraic graph transformation systems to a theory that is applicable to a wider range of categories satisfying suitable properties. The bases for HLR systems are adhesive categories.

On the other hand the Diagram Predicate Framework (DPF) provides a formalization of (meta-)modeling also considering diagrammatic constraints. DPF is an application oriented variant of the generalized sketch framework [1, 5] and can be considered as a natural extension of concepts used in graph transformation. In this report we analyze the present concept of specification in DPF, i.e., the concept of generalized sketch, and show that this concept does not provide adhesive categories. However, by introducing the concept of generalized DPF specification, i.e. of generalized multi sketches, we can repair this obstacle and obtain adhesive categories of DPF specifications. This will allow us to reuse results from graph transformations in the context of DPF and vice versa.

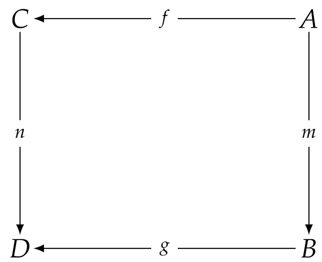
Since the difference between simple graphs and multi graphs shows the same pattern as the difference between generalized sketches and generalized multi sketches we start the report with a revised analysis of adhesiveness in the category of simple graphs and the category of multi graphs, respectively.

1 Introduction

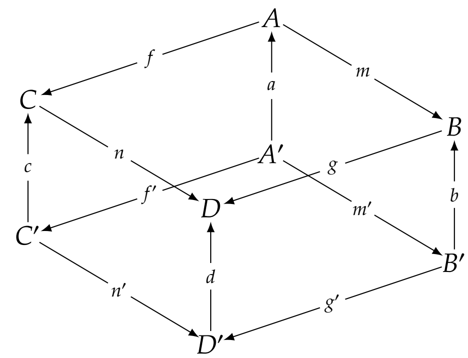
Adhesive categories [4] and adhesive HLR categories build a suitable categorical framework for graph transformation in the more general case. While adhesive categories are based on all monomorphisms in a category $\mathbf{C}$, adhesive HLR categories as well as weak adhesive HLR categories are based on a suitable subclass $\mathbf{M}$ of the monomorphisms in $\mathbf{C}$. The intuitive idea of adhesive categories is that of categories with suitable pushouts and pullbacks which are compatible with each other. Compatible is formalized by the concept of a van Kampen (VK) square. In this report we show that DPF, as it is defined up to now, does not provide adhesive categories. Then we will generalize DPF so that we get an adhesive category. This generalization will be analogue to the step from simple to multi graphs, graphs allowing more than one edge between a source and target vertex. Having this result we can reuse in our future work results from graph transformations like e.g. the Local Church-Rosser theorem, Parallelism Theorems, Embedding and Extension Theorems [3].

First we recall the definition of van Kampen squares and adhesive categories:

Definition 1.1 (van Kampen square)**.** *A pushout (1) is a van Kampen square if, for any commutative cube (2) with (1) in the top and where the back faces are pullbacks, the following statement holds: the bottom face is a pushout* *iff the front faces are pullbacks:*

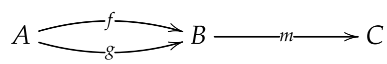


(1)



(2)

Definition 1.2 (Monomorphism). *Given a category* **C***, a morphism $m\colon B \to C$ is called a monomorphism if, for all morphisms $f, g : A \to B \in$* **C***, it holds that $f; m = g; m$ implies $f = g$:*



Definition 1.3 (adhesive category). *A category* **C** *is an adhesive category if:*

1. **C** *has pushouts along monomorphism (i.e. pushouts where at least one of the given morphisms is a monomorphism).*
2. **C** *has pullbacks.*
3. *Pushouts along monomorphisms are VK squares.*

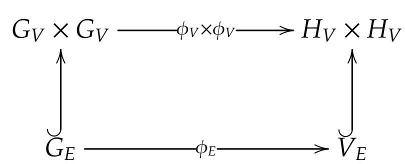
The central point in this definition is the Van Kampen property. Before we are going to show that categories of DPF specifications are not adhesive we inspect two categories of (directed) graphs.

2 Categories of Simple Directed Graphs

The category **Set** of sets and total mappings is a well-know adhesive category. The category of (simple) directed graphs **SGraph** extends the category of sets by a second component called "edges" that consists of a set of ordered pairs of vertices (*source*, *target*). We recall now the definition of directed graphs in the following. Note, that we call these graphs simple graphs here to distinguish them from multi graphs.

Definition 2.1 (Simple (directed) graph). *A (simple) directed graph $G = (G_V, G_E)$ consists of a set G_V of vertices (or nodes) and a set $G_E \subseteq G_V \times G_V$ of edges (or arrows) where each edge is an ordered pair of vertices $(x, y) \in G_V \times G_V$. The source respectively target of the edge is denoted by projection to the first ($src^G(x, y) = x$) respectively second ($trg^G(x, y) = y$) component of this pair.*

Definition 2.2 (Graph homomorphism in simple (directed) graphs). *A graph homomorphism $\phi : G \to H$ is a mapping $\phi_V : G_V \to H_V$ from the vertex set of G to the vertex set of H such that each edge $(x, y) \in G_E$ entails an edge $(\phi_V \times \phi_V)(x, y) := (\phi_V(x), \phi_V(y)) \in H_E$.*



Remark 2.1 (Mapping of edges). *The requirement $(\phi_V \times \phi_V)(G_E) \subseteq H_E$ means equivalently that the mapping $\phi_V \times \phi_V$ restricts to a mapping from G_E to H_E. We will denote this mapping by $\phi_E : G_E \to H_E$.*

Definition 2.3 (Category of simple (directed) graphs)**.** *The category* **SGraph** *has all simple (directed) graphs G as objects and all graph homomorphisms $\phi : G \to H$ as morphisms between graphs G and H. The associativity of composition of mappings ensures that the composition of two graph homomorphisms is a graph homomorphism as well and that the composition of graph homomorphisms is associative. Moreover, the identity mappings $id^{G_V} : G_V \to G_V$ define identity graph homomorphisms $id^G : G \to G$ and ensure that identity graph homomorphisms are left and right neutral with respect to composition.*

Fact 2.1 (Monomorphism in simple (directed) graphs)**.** *The monomorphisms in* **Set** *are exactly the injective mappings. In* **SGraph** *monomorphism are the morphism where the vertex mapping is injective. Note, that in this case also the induced mapping of edges is injective.*

2.1 Pushouts and Pullbacks of Simple Directed Graphs

Let us consider the general construction of pushouts and pullbacks in **SGraph** relying on the construction of pushouts and pullbacks in **Set**. This restricts the task mainly to the construction of edges for simple graphs.

Proposition 2.1 (Pushout for simple (directed) graphs)**.** *A pushout $B \xrightarrow{g^*} D \xleftarrow{f^*} C$ of a span $B \xleftarrow{f} A \xrightarrow{g} C$ of graph homomorphisms is obtained by constructing first a pushout $B_V \xrightarrow{g^*_V} D_V \xleftarrow{f^*_V} C_V$ in* **Set** *of the underlying span $B_V \xleftarrow{f_V} A_V \xrightarrow{g_V} C_V$ of mappings between sets of vertices and by defining the set of edges D_E as follows:*

$$D_E := (g^*_V \times g^*_V)(B_E) \cup (f^*_V \times f^*_V)(C_E) =$$
$$\{(g^*_V(x), g^*_V(y)) \mid (x, y) \in B_E\} \cup \{(f^*_V(x), f^*_V(y)) \mid (x, y) \in C_E\} \subseteq D_V \times D_V$$

Proof.

Homomorphism property: We have $(g^*_V \times g^*_V)(B_E) \subseteq D_E$ and $(f^*_V \times f^*_V)(C_E) \subseteq D_E$, by construction, thus the mappings g^*_V and f^*_V constitute graph homomorphisms $g^* : B \to D$ and $f^* : C \to D$, respectively.

Universal property:

1. There exists for all graph homomorphisms $g' : B \to X$ and $f' : C \to X$ with $g; f' = f; g'$, i. e. $g_V; f'_V = f_V; g'_V$, a unique mapping $k_V : D_V \to X_V$ with $f^*_V; k_V = f'_V$ and $g^*_V; k_V = g'_V$.
2. Since f^* and f' are graph homomorphisms we have

$$\begin{aligned}(k_V \times k_V)((f^*_V \times f^*_V)(C_E)) &= (f^*_V \times f^*_V); (k_V \times k_V)(C_E)\\ &= (f^*_V; k_V \times f^*_V; k_V)(C_E)\\ &= (f'_V \times f'_V)(C_E) \subseteq X_E\end{aligned}$$

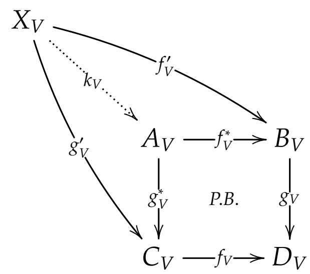
and, analogously, we have $(k_V \times k_V)((g^*_V \times g^*_V)(B_E)) \subseteq X_E$. Due to the construction of D_E this ensures $(k_V \times k_V)(D_E) \subseteq X_E$ thus k_V establishes a graph homomorphism $k : D \to X$.

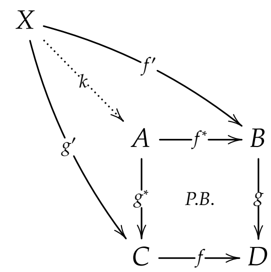
□

Besides pushouts, pullbacks in **SGraph** are also based on the corresponding construction in **Set**.

Proposition 2.2 (Pullback for simple (directed) graphs)**.** *A pullback $B \xleftarrow{f^*} A \xrightarrow{g^*} C$ of a co-span $B \xrightarrow{g} D \xleftarrow{f} C$ is obtained by constructing first a pullback $B_V \xleftarrow{f^*_V} A_V \xrightarrow{g^*_V} C_V$ in* **Set** *of the underlying co-span $B_V \xrightarrow{g_V} D_V \xleftarrow{f_V} C_V$ of mappings between sets of vertices and by defining the set of edges A_E as follows:*

$$A_E := \{e \in A_V \times A_V \mid (f^*_V \times f^*_V)(e) \in B_E, (g^*_V \times g^*_V)(e) \in C_E, g_E((f^*_V \times f^*_V)(e)) = f_E((g^*_V \times g^*_V)(e))\}$$





Proof.

Homomorphism property: By construction we have $(f^*_V \times f^*_V)(A_E) \subseteq B_E$ and $(g^*_V \times g^*_V)(C_E) \subseteq C_E$. Thus the mappings g^*_V and f^*_V constitute graph homomorphisms $f^* : A \to B$ and $g^* : A \to C$, respectively.

Universal property:

1. There exists for all graph homomorphisms $f' : E \to B$ and $g' : E \to C$ with $f'; g = g'; f$, i. e. $f'_V; g_V = g'_V; f_V$, a unique mapping $k_V : X_V \to A_V$ with $k_V; f^*_V = f'_V$ and $k_V; g^*_V = g'_V$.
2. Since f^* and f' are graph homomorphisms we have

$$\begin{aligned} (f^*_V \times f^*_V)((k_V \times k_V)(X_E)) \subseteq B_E &= (k_V \times k_V); (f^*_V \times f^*_V)(X_E) \\ &= (k_V; f^*_V \times k_V; f^*_V)(X_E) \\ &= (f'_V \times f'_V)(X_E) \end{aligned}$$

and, analogously, we have $(g^*_V \times g^*_V)((k_V \times k_V)(X_E)) \subseteq C_E$. Since we have $k_V; f^*_V; g_V = k_V; g^*_V; f_V$, by assumption, the construction of A_E ensures, in such a way, $(k_V \times k_V)(X_E) \subseteq A_E$ thus k_V establishes a graph homomorphism $k : X \to A$.

□

2.2 Van Kampen Property for Simple Graphs

It is a well known fact that the category of directed multi-graph **Graph** is adhesive [3]. It inherits adhesiveness from the category **Set**. But, what about the category **SGraph** of simple graphs? Let us have a look at the cube in Fig. 1. All homomorphisms are given by the identity mapping on the set of vertices and edges are mapped accordingly. Note, that these implicit mappings are unique since no multiple edges are allowed. The top and the bottom face of the cube in Fig. 1 are pushouts along monomorphism m respectively m', both back faces are pullbacks, hence both front faces of the cube have to be pullbacks to fulfill the Van Kampen property. However, while the left front face is trivially a valid pullback the right face is not. This means the category **SGraph** is not adhesive. The valid pullback according to the definition above is shown in Fig. 2.

What goes "wrong" in the category **SGraph**? Pullbacks are constructed by means of pullbacks, inverse images, intersection and equalizers in **Set**, i.e. by limit constructions, and should not, therefore, cause any problems. Pushouts in **SGraph** are based for vertices on pushouts in **Set** which is neither a problem. The construction of the set of edges, however, is based on the union of sets, which is not a colimit construction in **Set**. This union construction in pushouts causes the problem since edges collapse in the pushout object, i.e. they cannot be traced back. If there is an edge in the pushout we cannot find out if this edge originates from left or from right or from both sides. To have such a tracing back facility, we have to move from simple to multi graphs.

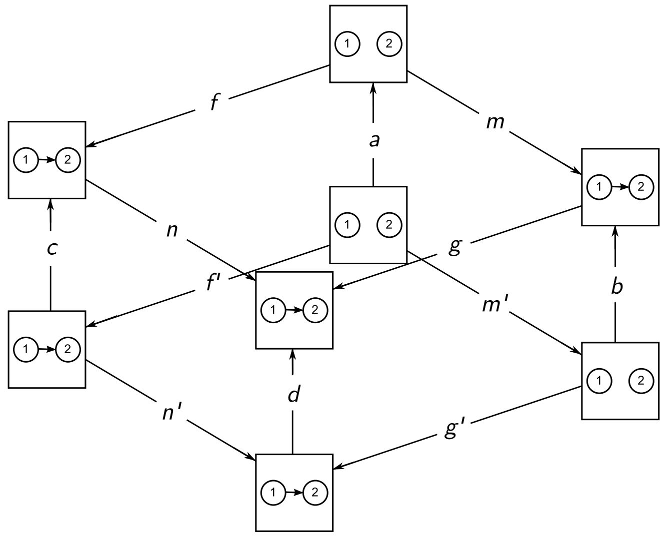


Figure 1: Counter example Van Kampen property in **SGraph**

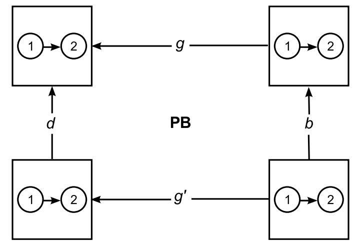


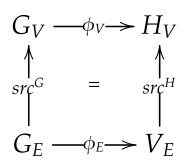
Figure 2: Corrected pullback in **SGraph**

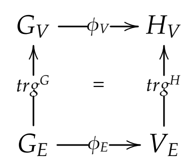
3 Category of (Directed Multi) Graphs

First let us recall the basic definitions for multi graphs.

Definition 3.1 ((Directed multi) graph)**.** *A multi graph $G = (G_V, G_E, src^G, trg^G)$ consists of a set G_V of vertices (or nodes), a set G_E of edges (or arrows) and two maps $src^G, trg^G : G_E \to G_V$ assigning the source and target to each edge, respectively. $e : x \to y$ denotes that $src^G(e) = x$ and $trg^G(e) = y$.*

Definition 3.2 (Graph homomorphism in (directed multi) graphs)**.** *A graph homomorphism $\phi : G \to H$ consists of a pair of maps $\phi_V : G_V \to H_V$, $\phi_E : G_E \to H_E$ which preserve the graph structure, i.e., for each edge $e : x \to y$ in G we have $\phi_E(e) : \phi_V(x) \to \phi_V(y)$ in H, i.e., $src^G; \phi_V = \phi_E; src^H$ and $trg^G; \phi_V = \phi_E; trg^H$.*





Definition 3.3 (Category of (directed multi) graphs)**.** *The category* **Graph** *has all multi graphs G as objects and all graph homomorphisms $\phi : G \to H$ as morphisms.*

The composition $\phi;\psi : G \to K$ of two graph homomorphisms $\phi : G \to H$ and $\psi : H \to K$ is defined component-wise $\phi;\psi = (\phi_V, \phi_E);(\psi_V, \psi_E) := (\phi_V;\psi_V, \phi_E;\psi_E)$. The identity graph homomorphisms $id^G : G \to G$ are also defined component-wise $id^G = (id^{G_V}, id^{G_E})$. This ensures that the composition of graph homomorphisms is associative and that identity graph homomorphisms are left and right neutral with respect to composition.

Based on general results about functor categories it is shown in [3] that pushouts, pullbacks, epimorphisms, and monomorphisms in **Graph** are exactly given by component-wise pushouts, pullbacks, epimorphisms, and monomorphisms, respectively, in **Set**. We develop here these results in a more basic, systematic and detailed way to prepare, in an appropriate way, our investigations of categories of diagrammatic specifications.

3.1 Pushouts and Pullbacks of Directed Multi Graphs

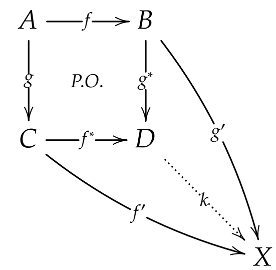
In the following we show that pushouts respectively pullbacks in the category **Graph** are indeed given by pushouts respectively pullbacks in the underlying category **Set**.

Proposition 3.1 (Pushout of (directed multi) graphs)**.** *A pushout $B \xrightarrow{g^*} D \xleftarrow{f^*} C$ of a span $B \xleftarrow{f} A \xrightarrow{g} C$ of graph homomorphisms is obtained by constructing component-wise a pushout in* **Set** *for the underlying* *maps between sets of vertices and sets of edges, respectively.* $src^D : D_E \to D_V$ *and* $trg^D : D_E \to D_V$ *are the unique mediating maps such that* ($g_E^*; src^D = src^B; g_V^*$ *and* $f_E^*; src^D = src^B; f_V^*$) *or* ($g_E^*; trg^D = trg^B; g_V^*$ *and* $f_E^*; trg^D = trg^B; f_V^*$)*, respectively.*

That is, due to the construction of pushouts in **Set** *for each edge* $e : x \to y \in D_E$ *the* $src^D(e)$ *respectively* $trg^D(e)$ *is given by:*

$$src^D(e) = \begin{cases} g_V^*(src^B(e')) & \text{if } \exists e' \in B_E \text{ with } g_E^*(e') = e \\ f_V^*(src^C(e')) & \text{else } \exists e' \in C_E \text{ with } f_E^*(e') = e \end{cases}$$

$$trg^D(e) = \begin{cases} g_V^*(trg^B(e')) & \text{if } \exists e' \in B_E \text{ with } g_E^*(e') = e \\ f_V^*(trg^C(e')) & \text{else } \exists e' \in C_E \text{ with } f_E^*(e') = e \end{cases}$$

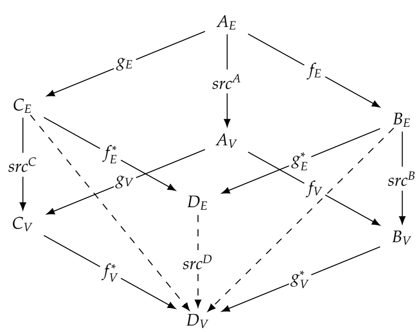


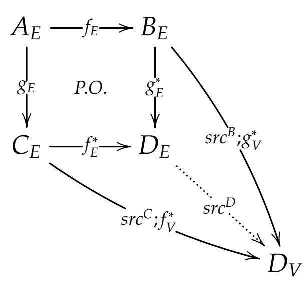
Proof. **"⇐": Componentwise pushouts in Set provide pushouts in Graph**

Existence of source and target maps and homomorphism property: The top face of the cube in the figure below shows the pushout for edges in **Set** while the pushout for vertices in **Set** is shown in its bottom face. We have

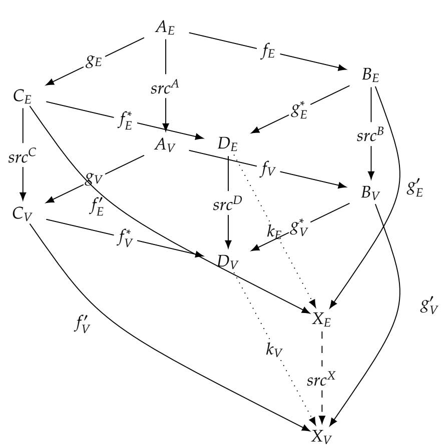
$g_E; src^C; f_V^*$	$=$	$src^A; g_V; f_V^*$	g graph homomorphism
	$=$	$src^A; f_V; g_V^*$	bottom face is commutative
	$=$	$f_E; src^B; g_V^*$	f graph homom.

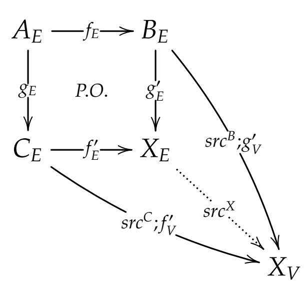
And, analogously, $g_E; trg^C; f_V^* = f_E; trg^B; g_V^*$ thus the uniqueness of mediating morphisms for the pushout for edges entails indeed the existence of unique mappings src^D respectively trg^D satisfying the equations above. The validity of these equations means, at the same time, that (g_V^*, g_E^*) defines a graph homomorphism $g^* : B \to D$ and that (f_V^*, f_E^*) defines a graph homomorphism $f^* : C \to D$, respectively.





Universal property: Assume graph X and two graph homomorphisms f' and g' so that $g; f' = f; g'$ (see figures below).





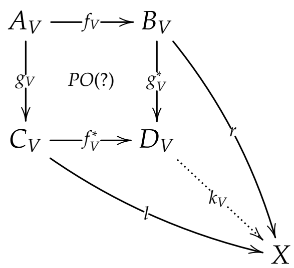
1. There are unique mediating mappings k_E and k_V since top and bottom face of the cube are pushouts.

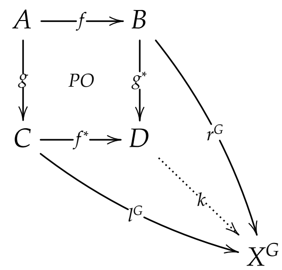
2. By the construction of (k_V, k_E) and the homomorphism property of f', f^*, g' and g^*, respectively, we get $f^*_E;k_E;src^X = f^*_E;src^D;k_V$, and $g^*_E;k_E;src^X = g^*_E;src^D;k_V$, and $f^*_E;k_E;trg^X = f^*_E;trg^D;k_V$, and $g^*_E;k_E;trg^X = g^*_E;trg^D;k_V$ thus uniqueness of mediating morphisms for the top pushout entails $k_E;src^X = src^D;k_V$ and $k_E;trg^X = trg^D;k_V$. That is (k_V, k_E) defines indeed a graph homomorphism $k : D \to X$.

"⇒": Pushouts in Graph imply componentwise pushouts in Set

Let a pushout $B \xrightarrow{g^*} D \xleftarrow{f^*} C$ of a span $B \xleftarrow{f} A \xrightarrow{g} C$ of graph homomorphisms be given.

Pushout for vertices: We consider a set X and a maps $l : C_V \to X$, $r : B_V \to X$ such that $g_V;l = f_V;r$.





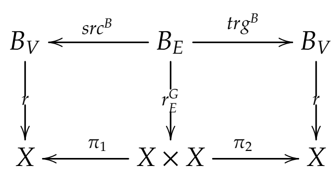
Existence of mediating map: We construct a graph X^G as follows:

$$X^G_V := X,\ X^G_E := X \times X,\ src^{X^G} := \pi_1,\ trg^{X^G} := \pi_2.$$

The map r can be extended then to a graph homomorphism $r^G : B \to X^G$:

$$r^G_V := r,\ r^G_E := \langle src^B;r, trg^B;r\rangle,$$

that is, r^G_E is the unique map such that the following diagram commutes



This ensures also that (r^G_V, r^G_E) defines a graph homomorphism. $l^G : C \to X^G$ is defined analogously.

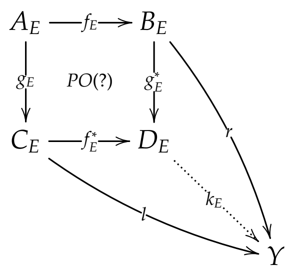
By construction and assumption we have $f_V;r^G_V = g_V;l^G_V$ and further

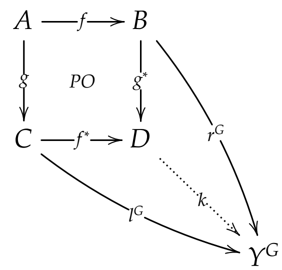
$f_E;r^G_E$	$=$	$f_E;\langle src^B;r, trg^B;r\rangle$	definition r^G_E
	$=$	$\langle f_E;src^B;r, f_E;trg^B;r\rangle$	pre-composition with tuples
	$=$	$\langle src^A;f_V;r, trg^A;f_V;r\rangle$	f graph homom.
	$=$	$\langle src^A;g_V;l, trg^A;g_V;l\rangle$	assumption
	$=$	$\langle g_E;src^C;l, g_E;trg^C;l\rangle$	g graph homom.
	$=$	$g_E;\langle src^C;l, trg^C;l\rangle$	pre-composition with tuples
	$=$	$g_E;l^G_E$	definition l^G_E

In such a way, we have $f;r^G = g;l^G$ in **Graph** thus there exists a unique graph homomorphism $k : D \to X^G$ such that $g^*;k = r^G$ and $f^*;k = l^G$. Especially, we have a mediating map $k_V : D_V \to X$ such that $g^*_V;k_V = r^G_V = r$ and $f^*_V;k_V = l^G_V = l$.

Uniqueness of mediating map: For any $\bar{k} : D_V \to X$ such that $g^*_V;\bar{k} = r$ and $f^*_V;\bar{k} = l$ we get $g^*;\bar{k}^G = r^G$ and $f^*;\bar{k}^G = l^G$ according to our constructions, thus the uniqueness of mediators in **Graph** implies $k = \bar{k}^G$ and, especially, $k_V = \bar{k}^G_V = \bar{k}$.

Pushout for edges: We consider a set Y and a maps $l : C_V \to Y$, $r : B_V \to Y$ such that $g_E; l = f_E; r$.



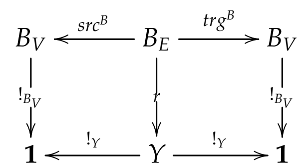


Existence of mediating map: We construct a graph Y^G as follows:

$$Y^G_V := \mathbf{1},\ Y^G_E := Y,\ src^{Y^G} = trg^{Y^G} :=!_Y : Y \to \mathbf{1}$$

where $\mathbf{1}$ is a singleton set, i.e., a terminal object in **Set**. The map r can be extended then to a graph homomorphism $r^G : B \to Y^G$:

$$r^G_V :=!_{B_V},\ r^G_E := r.$$



By uniqueness of terminal maps this ensures that (r^G_V, r^G_E) defines a graph homomorphism. $l^G : C \to X^G$ is defined analogously.

By construction and assumption we have $f_E; r^G_E = g_E; l^G_E$ and further $f_V; r^G_V = g_V; l^G_V$ again due to the uniqueness of terminal maps. This gives $f; r^G = g; l^G$ in **Graph** thus there exists a unique graph homomorphism $k : D \to Y^G$ such that $g^*; k = r^G$ and $f^*; k = l^G$. Especially, we have a mediating map $k_E : D_E \to Y$ such that $g^*_E; k_E = r^G_E = r$ and $f^*_E; k_E = l^G_E = l$.

Uniqueness of mediating map: Analogously to the case of pushouts for vertices.

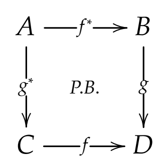
□

Now we consider pullbacks.

Proposition 3.2 (Pullback of (directed multi) graphs)**.** *A pullback $B \xleftarrow{f^*} A \xrightarrow{g^*} C$ of a co-span $B \xrightarrow{g} D \xleftarrow{f} C$ is obtained by constructing component-wise a pullback in* **Set** *for the underlying maps between sets of vertices and sets of edges, respectively. $src^A : A_E \to A_V$ and $trg^A : A_E \to A_V$ are the unique mediating maps such that ($src^A; g^*_V = g^*_E; src^C$ and $src^A; f^*_V = f^*_E; src^B$) or ($trg^A; g^*_V = g^*_E; trg^C$ and $trg^A; f^*_V = f^*_E; trg^B$), respectively.*

That is, if we construct pullbacks in **Set** *by Cartesian products and equalizer, we have for each edge $e = (e_C, e_B) \in A_E$:*

$$src^A(e_C, e_B) := (src^C(e_C), src^B(e_B))$$
$$trg^A(e_C, e_B) := (trg^C(e_C), trg^B(e_B)).$$

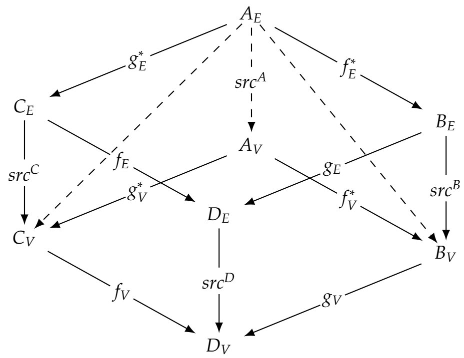


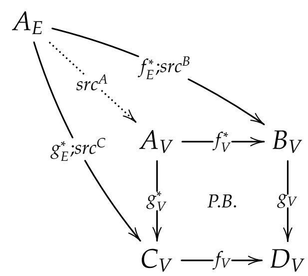
Proof. **"⇐": Componentwise pullbacks in Set provide pullbacks in Graph**

Existence of source and target maps and homomorphism property: The top face of the cube in the figure below shows the pullback for edges in **Set** while the pullback for vertices in **Set** is shown in its bottom face. We have

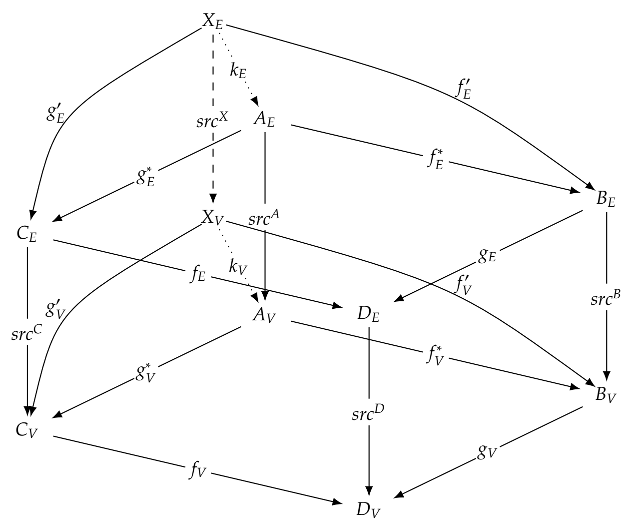
$g^*_E; src^C; f_V$	$=$	$g^*_E; f_E; src^D$	f graph homomorphism
	$=$	$f^*_E; g_E; src^D$	bottom face is commutative
	$=$	$f^*_E; src^B; g_V$	g graph homom.

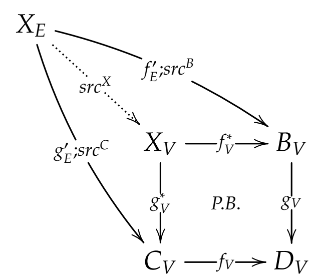
and, analogously, $g^*_E; trg^C; f_V = f^*_E; trg^B; g^*_V$ thus the uniqueness of mediating morphisms for the pullback for vertices entails indeed the existence of unique mappings src^D respectively trg^D satisfying the equations above. The validity of these equations means, at the same time, that (g^*_V, g^*_E) defines a graph homomorphism $g^* : A \to C$ and that (f^*_V, f^*_E) defines a graph homomorphism $f^* : A \to B$, respectively.





Universal property: Assume graph X and two graph homomorphisms f' and g' so that $g'; f = f'; g$ (see figures below).





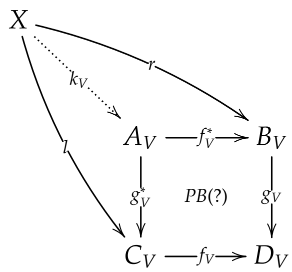
1. There are unique mappings k_E and k_V since top and bottom face of the cube are pullbacks.

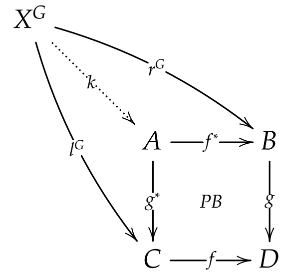
2. By the construction of (k_V, k_E) and the homomorphism property of f', f^*, g' and g^*, respectively, we get $k_E; src^A; f^*_V = src^X; k_V; f^*_V$, and $k_E; src^A; g^*_V = src^X; k_V; g^*_V$, and $k_E; trg^A; f^*_V = trg^X; k_V; f^*_V$, and $k_E; trg^A; g^*_V = trg^X; k_V; g^*_V$, thus uniqueness of mediating morphisms for the bottom pullback entails $k_E; src^A = src^X; k_V$ and $k_E; trg^A = trg^X; k_V$. That is (k_V, k_E) defines indeed a graph homomorphism $k : X \to A$.

"⇒": Pullbacks in Graph imply pullbacks of their components in Set

Let a pullback $B \xleftarrow{f^*} A \xrightarrow{g^*} C$ of a co-span $B \xrightarrow{g} D \xleftarrow{f} C$ of graph homomorphisms be given.

Pullbacks for vertices: We consider a set X and maps $l : X \to C_V$, $r : X \to B_V$ such that $l; g_V = r; f_V$.



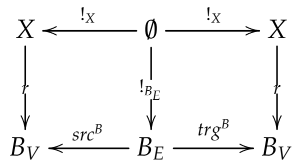


Existence of mediating map: We construct a graph X^G as follows:

$$X^G_V := X,\ X^G_E := \emptyset,\ src^{X^G} = trg^{X^G} := !_X : \emptyset \to X,$$

where $\emptyset$ is the empty set, i.e. the initial object in **Set**. The map r can be extended then to a graph homomorphism $r^G : X^G \to B$:

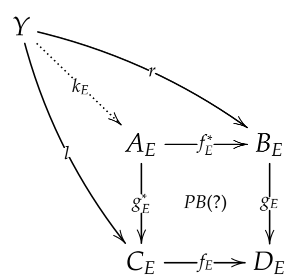
$$r^G_V := r,\ r^G_E := !_{B_E}.$$

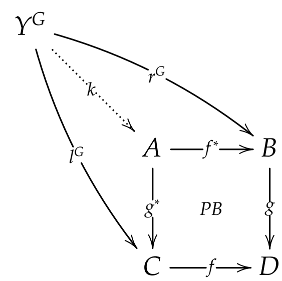


By uniqueness of initial maps this ensures that (r^G_V, r^G_E) defines a graph homomorphism $r^G : X^G \to B$. $l^G : X^G \to C$ is defined analogously. We have $l^G_V; f_V = r^G_V; g_V$ by assumption and $l^G_E; f_E = r^G_E; g_E$ by uniqueness of initial maps. This means $l^G; f = r^G; g$ in **Graph** thus there exists a unique graph homomorphism $k : X^G \to A$ such that $k; g^* = l^G$ and $k; f^* = r^G$. Especially, we have $k_V; g^*_V = l^G_V$ and $k_V; f^*_V = r^G_V = r$, as desired.

Uniqueness of mediating map: Analogously to the case of pushouts for vertices.

Pullbacks for edges: We consider a set Y and maps $l : Y \to C_E$, $r : Y \to B_E$ such that $l; g_E = r; f_E$.





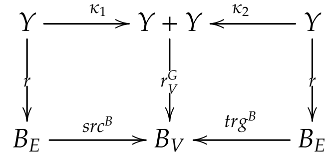
Existence of mediating map: We construct a graph Y^G as follows:

$$Y^G_V := Y + Y,\ Y^G_E := Y,\ src^{Y^G} := \kappa_1,\ trg^{Y^G} := \kappa_2.$$

The map r can be extended then to a graph homomorphism $r^G : Y^G \to B$:

$$r^G_V := [r; src^B, r; trg^B],\ r^G_E := r,$$

i.e. r^G_V is the unique map such that the following diagram commutes



This ensures also that (r^G_V, r^G_E) defines a graph homomorphism $r^G : Y^G \to B$. $l^G : Y^G \to C$ is defined analogously. By construction and assumption we have $l^G_E; f_E = r^G_E; g_E$ and, dually to the case of pushouts for nodes, we can show $l^G_V; f_V = r^G_V; g_V$. This means $l^G; f = r^G; g$ in **Graph** thus there exists a unique graph homomorphism $k : Y^G \to A$ such that $k; g^* = l^G$ and $k; f^* = r^G$. Especially, we have $k_V; g^*_V = l^G_V$ and $k_V; f^*_V = r^G_V = r$, as desired.

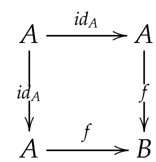
Uniqueness of mediating map: Analogously to the case of pushouts for vertices.

□

3.2 Van Kampen Property for Directed Multi Graphs

In the following, we consider the VK property for **Graph**. Since, this time the construction of pushouts does not rely on a union construction anymore, we expect the category to be adhesive.

Lemma 3.1 (Monic = Pullback)**.**
A morphism $f : A \to B$ in a category **C** *is a monomorphism iff the figure on the right is a pullback diagram.*



Corollary 3.1 (Monomorphism in (directed multi) graphs)**.** *Monomorphisms in* **Graph** *are given by componentwise monomorphisms in* **Set** *i.e. a graph homomorphism $f : A \to B$ is a monomorphism iff $f_V : A_V \to B_V$ and $f_E : A_E \to B_E$ are monomorphisms.*

Proof. Since the identities in **Graph** are componentwise identities we get by Lemma 3.1, Proposition 3.2 and again Lemma 3.1 for any graph homomorphism in $f : A \to B$:

f is a monomorphism in **Graph**

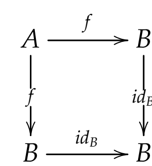
$\iff$ $A \xleftarrow{id_A} A \xrightarrow{id_A} A$ is the pullback of $A \xrightarrow{f} B \xleftarrow{f} A$

$\iff$ $A_V \xleftarrow{id_{AV}} A_V \xrightarrow{id_{AV}} A_V$ is the pullback of $A_V \xrightarrow{f_V} B_V \xleftarrow{f_V} A_V$ and
$A_E \xleftarrow{id_{AE}} A_E \xrightarrow{id_{AE}} A_E$ is the pullback of $A_E \xrightarrow{f_E} B_E \xleftarrow{f_E} A_E$ in **Set**

$\iff$ $f_V : A_V \to B_V$ and $f_E : A_E \to B_E$ are monomorphism in **Set**

□

Lemma 3.2 (Epic = Pushout)**.**
A morphism $f : A \to B$ in a Category **C** *is an epimorphism iff the figure on the right is a pushout diagram.*



Corollary 3.2 (Epimorphism in (directed multi) graphs)**.** *Epimorphisms in* **Graph** *are given by componentwise epimorphisms in* **Set** *i.e. a graph homomorphism $f : G \to H$ is an epimorphism iff $f_V : G_V \to H_V$ and $f_E : G_E \to H_E$ are epimorphisms.*

Proof. Since the identities in **Graph** are componentwise identities we get by Lemma 3.2, Proposition 3.1 and again Lemma 3.2 for any graph homomorphism in $f : A \to B$:

f is an epimorphism in **Graph**

$\iff$ $B \xrightarrow{id_B} B \xleftarrow{id_B} B$ is the pushout of $B \xleftarrow{f} A \xrightarrow{f} B$

$\iff$ $B_V \xrightarrow{id_{BV}} B_V \xleftarrow{id_{BV}} B_V$ is the pushout of $B_V \xleftarrow{f_V} A_V \xrightarrow{f_V} B_V$ and
$B_E \xrightarrow{id_{BE}} B_E \xleftarrow{id_{BE}} B_E$ is the pushout of $B_E \xleftarrow{f_E} A_E \xrightarrow{f_E} B_E$

$\iff$ $f_V : A_V \to B_V$ and $f_E : A_E \to B_E$ are epimorphism in **Set**

□

Considering all facts about the category of directed multi-graph **Graph**, we can conclude that it is adhesive [3] and that it inherits adhesiveness from category **Set**.

Proposition 3.3. *Pushouts along monomorphism are VK squares in* **Graph***.*

Proof. Assume a commutative cube (2) in **Graph** (See Definition 1.1). Since composition of graph homomorphism is defined componentwise we have (2) commutes in **Graph** iff the corresponding two cubes for vertices and edges commute in **Set**. We assume top face in (2) is a pushout and both back faces in (2) are pullbacks. By Proposition 3.1 and Proposition 3.2 follows:

$$\begin{aligned} \text{top face in (1) is a pushout} &\iff \text{top face in (1) for } (1_V) \text{ and } (1_E) \text{ are pushouts and} \\ \text{back faces in (2) are pullbacks} &\iff \text{back faces in (2) for } (2_V) \text{ and } (2_E) \text{ are pullbacks} \end{aligned}$$

We have by Proposition 3.1, Proposition 3.2 and **Set** is adhesive:

$$\begin{aligned} \text{bottom face in (2) is pushout} &\iff \text{bottom faces in } (2_V) \text{ and } (2_E) \text{ are pushouts} \\ &\iff \text{front faces in } (2_V) \text{ and } (2_E) \text{ are pullbacks} \\ &\iff \text{front faces in (2) are pullbacks} \end{aligned}$$

□

Let us review Figure 1 which shows a counterexample for the VK property in **SGraph**. The cube in the figure is not a valid VK cube even top and bottom faces are pushouts and both back faces are pullbacks. Figure 3 shows the analog example for **Graph**. This time we also have to map the edges since we can have more than one edge having the same source and target vertex. In contrast to the earlier example we get two edges in the upper pushout graph. Therefore, we can correctly trace back the edges in the right front face of the cube which shows in contrast to Figure 1 a valid pullback diagram. Hence Figure 3 shows a valid VK cube.

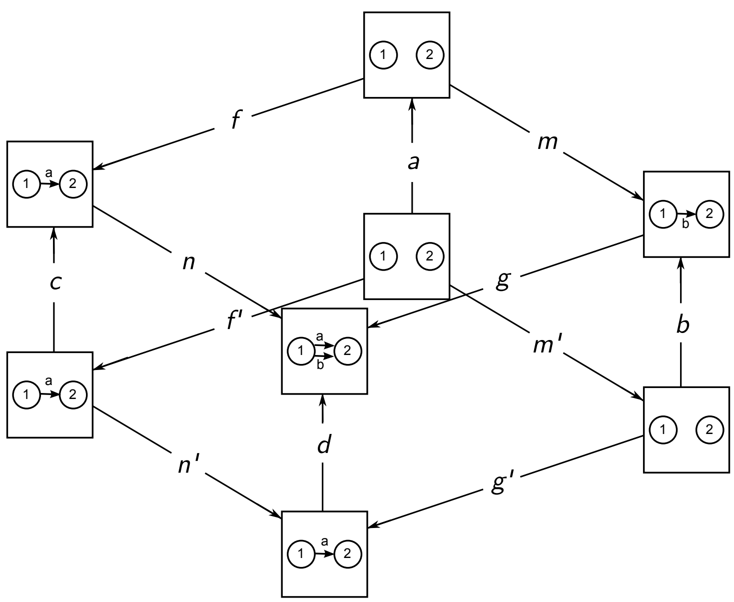


Figure 3: Example Van Kampen property in **Graph**

4 Category of DPF Specifications

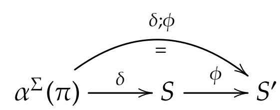
Now, we consider **Spec**, the category of DPF specifications, and show that it is not adhesive for a similar reason **SGraph** is not adhesive. Therefore, let us recall the main definitions from [7] and [6] in the following:

Definition 4.1 (Signature). *A signature $\Sigma = (\Pi^\Sigma, \alpha^\Sigma)$ consists of a set of predicate symbols Π^Σ and a mapping α^Σ which assigns a (multi) graph to each predicate symbol $\pi \in \Pi^\Sigma$. $\alpha^\Sigma(\pi)$ is called the* arity *of the predicate symbol π.*

Definition 4.2 (Atomic constraint). *Given a signature $\Sigma = (\Pi^\Sigma, \alpha^\Sigma)$, an atomic constraint (π, δ) on a (multi) graph S consists of a predicate symbol $\pi \in \Pi^\Sigma$ and a graph homomorphism $\delta : \alpha^\Sigma(\pi) \to S$.*

Definition 4.3 (Specification). *Given a signature $\Sigma = (\Pi^\Sigma, \alpha^\Sigma)$, a specification $\mathfrak{S} = (S, C^{\mathfrak{S}} \colon \Sigma)$ consists of a multi graph S and a set $C^{\mathfrak{S}}$ of atomic constraints (π, δ) on S with $\pi \in \Pi^\Sigma$.*

Definition 4.4 (Specification morphism). *Given two specifications $\mathfrak{S} = (S, C^{\mathfrak{S}} \colon \Sigma)$ and $\mathfrak{S}' = (S', C^{\mathfrak{S}'} \colon \Sigma)$, a specification morphism $\phi : \mathfrak{S} \to \mathfrak{S}'$ is a graph homomorphism $\phi : S \to S'$ such that $(\pi, \delta) \in C^{\mathfrak{S}}$ implies $\phi_C(\pi, \delta) := (\pi, \delta; \phi) \in C^{\mathfrak{S}'}$.*



Definition 4.5 (Category of specifications). *Given a signature $\Sigma = (\Pi^\Sigma, \alpha^\Sigma)$, the category* **Spec**$(\Sigma)$ *has all specifications $\mathfrak{S} = (S, C^{\mathfrak{S}} \colon \Sigma)$ as objects and all specification morphisms $\phi : \mathfrak{S} \to \mathfrak{S}'$ as morphisms between specifications $\mathfrak{S}$ and $\mathfrak{S}'$.*

The associativity of composition of graph homomorphism ensures that the composition of two specification morphisms is a specification morphism as well and that the composition of specification morphisms is associative. Moreover, the identity graph homomorphisms $id_S : S \to S$ define identity specification morphisms $id_{\mathfrak{S}} : \mathfrak{S} \to \mathfrak{S}$ and ensure that identity specification morphisms are left and right neutral with respect to composition.

Remark 4.1 (Monomorphism for specifications). *Monomorphism in* **Spec**(Σ) *are the morphism where the underlying graph homomorphism is a monomorphism. Definition 4.4 ensures that the translation $\phi_C : C^{\mathfrak{S}} \to C^{\mathfrak{S}'}$ of atomic constraints becomes, in this case, also an injective mapping. Compare Remark 2.1, where we have the same effect for the translation of edges.*

4.1 Pushouts and Pullbacks of DPF Specifications

In this section we will consider the general construction of pushouts and pullbacks in **Spec** relying on the existence of pushouts and pullbacks in **Graph** as done in [8] for generalized sketches[1]. This restricts the task mainly on the consideration of atomic constraints in the context of **Spec**.

Note, the category of (typed) **conformant** specifications [6, 7] will not be considered here since **conformance** rely on arbitrary semantics assigned to atomic constraints which are in general not specified in terms of graph homomorphism and commuting diagrams.

Similar to pushouts in **SGraph** pushouts in **Spec** are based on pushouts in **Graph** and the union of sets (of constraints).

Proposition 4.1 (Pushout of specifications). *A pushout $\mathfrak{D} = (D, C^{\mathfrak{D}} \colon \Sigma)$ of a span $\mathfrak{B} \xleftarrow{f} \mathfrak{A} \xrightarrow{g} \mathfrak{C}$ of a specification morphisms is obtained by constructing a pushout in* **Graph** *for the underlying graph homomorphisms and by defining the set of atomic constraints $C^{\mathfrak{D}}$ as follows:*

$$C^{\mathfrak{D}} := \{(\pi, \delta; g^*) \mid (\pi, \delta) \in C^{\mathfrak{B}}\} \cup \{(\pi, \delta; f^*) \mid (\pi, \delta) \in C^{\mathfrak{C}}\}$$

[1] The construction for pushouts and pullbacks in **Spec** can also be found in [7] and [6] for restricted cases.

Proof.

Morphism property: We have $f^*_C(C^{\mathfrak{C}}) \subseteq C^{\mathfrak{D}}$ and $g^*_C(C^{\mathfrak{B}}) \subseteq C^{\mathfrak{D}}$, by construction, thus the graph homomorphisms f^* and g^* constitute specification morphisms $f^* : \mathfrak{C} \to \mathfrak{D}$ and $g^* : \mathfrak{B} \to \mathfrak{D}$, respectively.

Universal property:

1. There exists for all specification morphisms $g' : \mathfrak{B} \to \mathfrak{X}$ and $f' : \mathfrak{C} \to \mathfrak{X}$ with $g; f' = f; g'$ i.e. $g; f' = f; g'$ a unique graph homomorphism $k : D \to X$ with $f^*; k = f'$ and $g^*; k = g'$.
2. Since f^* and f' are specification morphisms we have for any $\pi \in \Sigma$ and $(\pi, \delta_C) \in C^{\mathfrak{C}}$

$$(\pi, \delta_C; f') \quad = \quad (\pi, \delta_C; f^*; k) \subseteq C^{\mathfrak{X}}$$

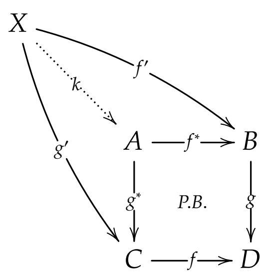
and analogously $(\pi, \delta_B; g^*; k) \subseteq C^{\mathfrak{X}}$ for any $\pi \in \Sigma$ and $(\pi, \delta_B) \in C^{\mathfrak{B}}$. Due to the construction of $C^{\mathfrak{D}}$ this ensures $(\pi, \delta_D; k) \subseteq C^{\mathfrak{X}}$, for all $(\pi, \delta_D) \in C^{\mathfrak{D}}$, thus k establishes a specification morphism $k : \mathfrak{D} \to \mathfrak{X}$.

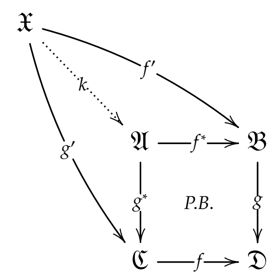
□

Similar to pullbacks in **SGraph** pullbacks in **Spec** rely as well on pullbacks in **Graph** as on intersection and equalizers in **Set**.

Proposition 4.2 (Pullback of specifications)**.** *A pullback $\mathfrak{A} = (A, C^{\mathfrak{A}} : \Sigma)$ of a co-span $B \xrightarrow{g} D \xleftarrow{f} C$ is obtained by constructing a pullback in* **Graph** *for the underlying graph homomorphisms and by defining the set of atomic constraints $C^{\mathfrak{A}}$ as follows:*

$$C^{\mathfrak{A}} := \{(\pi, \delta : \alpha^{\Sigma}(\pi) \to A) \mid (\pi, \delta; f^*) \in C^{\mathfrak{B}} \text{ and } (\pi, \delta; g^*) \in C^{\mathfrak{C}} \text{ and } (\pi, \delta; f^*; g) = (\pi, \delta; g^*; f) \in C^{\mathfrak{D}}\}$$





Proof.

Morphism property: By construction, we have $f^*_C(C^{\mathfrak{A}}) \subseteq C^{\mathfrak{B}}$ and $g^*_C(C^{\mathfrak{A}}) \subseteq C^{\mathfrak{C}}$ thus the graph homomorphisms f^* and g^* constitute specification morphisms $f^* : \mathfrak{A} \to \mathfrak{B}$ and $g^* : \mathfrak{A} \to \mathfrak{C}$, respectively.

Universal property:

1. There exists for all specification morphisms $f' : \mathfrak{X} \to \mathfrak{B}$ and $g' : \mathfrak{X} \to \mathfrak{C}$ with $f'; g = g'; f$ a unique graph homomorphism $k : X \to A$ with $k; f^* = f'$ and $k; g^* = g'$.
2. Since f^* and f' are specification morphisms we have for any $\pi \in \Sigma$ and $(\pi, \delta_X) \in C^{\mathfrak{X}}$

$$(\pi, \delta_X; f') \quad = \quad (\pi, \delta_X; k; f^*) \subseteq C^{\mathfrak{B}}$$

and analogously $(\pi, \delta_X; k; g^*) \subseteq C^{\mathfrak{C}}$ for any $\pi \in \Sigma$ and $(\pi, \delta_X) \in C^{\mathfrak{X}}$. Since we have $k; f^*; g = k; g^*; f$, by assumption, the construction of $C^{\mathfrak{A}}$ ensures, in such a way, $(\pi, \delta_X; k) \subseteq C^{\mathfrak{A}}$ for any $\pi \in \Sigma$ and $(\pi, \delta_X) \in C^{\mathfrak{X}}$ thus k establishes a specification morphism $k : \mathfrak{X} \to \mathfrak{A}$.

□

4.2 Van Kampen Property for DPF Specifications

In Section 2 we considered **SGraph** and in particularly pushouts in **SGraph**. In Subsection 2.2, we have presented a counterexample for adhesiveness in **SGraph** and realized that **SGraph** is not adhesive due to the fact that pushouts rely on a union construction on edges.

Now, we analyze pushouts in **Spec**. Since pushouts in **Spec** rely again on a union construction, we expect the category **Spec** of specifications not to be adhesive. And indeed **Spec** is not adhesive. Figure 4 shows a counterexample for **Spec**. Predicates are given as concrete sets. The top and the bottom face are pushouts along monomorphisms. The back faces are pullbacks. Also the left front face is a pullback. However, the right front face is not a pullback since $\pi_1 \in C^{\mathfrak{B}}$ and $\pi_1 \in C^{\mathfrak{D}'}$ but $\pi_1 \notin C^{\mathfrak{B}'}$.

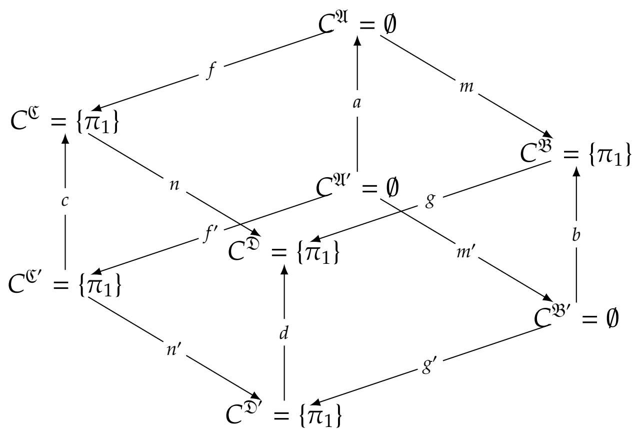


Figure 4: Counter example Van Kampen property in **Spec**

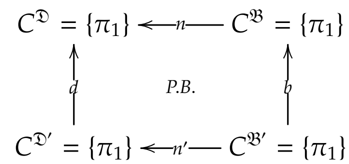


Figure 5: Corrected pullback in **Spec**

Note, that the reason is exactly the same as before in **SGraph**. Pushouts in **Spec** rely on a union construction and hence predicates cannot be traced back. The valid pullback according to the definition above is shown in Fig. 5.

In contrast to **SGraph** we have seen that category **Graph** is adhesive and that the main difference between both is that the pushout construction in **Graph** only rely on pushouts in **Set**. In the following, we will consider the category of generalized DPF Specifications which "repairs" the category of "usual" DPF specification so that we get an adhesive one.

5 The Category of Generalized DPF Specifications

In the following we analyze the definitions of Section 4 to prepare for a generalization[2].

To lift the definitions in Section 4 to a more structured and abstract level, we make, first, explicit the arity mapping $\alpha^\Sigma : \Pi^\Sigma \to \mathbf{Graph}_0$ where $\mathbf{Graph}_0$ denotes the set of objects in the category **Graph**. Second, we consider the slice category $(\mathbf{Graph}/S)$ and the mapping $fst^S : (\mathbf{Graph}/S)_0 \to \mathbf{Graph}_0$ assigning to each object $\varphi : G \to S$ in $(\mathbf{Graph}/S)$ the domain G. Then we consider the pullback of the co-span

$$\Pi^\Sigma \xrightarrow{\alpha^\Sigma} \mathbf{Graph}_0 \xleftarrow{fst^S} (\mathbf{Graph}/S)_0,$$

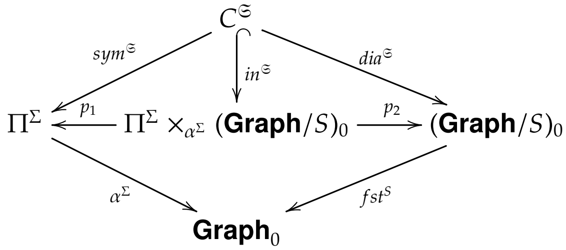
i.e., the set

$$\Pi^\Sigma \times_{\alpha^\Sigma} (\mathbf{Graph}/S)_0 = \{(\pi, \varphi) \in \Pi^\Sigma \times (\mathbf{Graph}/S)_0 \mid \alpha^\Sigma(\pi) = fst^S(\varphi)\}.$$

The main point is that our "traditional" atomic constraints from Definition 4.2 are exactly the elements of $\Pi^\Sigma \times_{\alpha^\Sigma} (\mathbf{Graph}/S)_0$. That is, a "traditional" specification is given by a multi graph S and a subset

[2]The formalization of this section is based on insights and ideas of Zinovy Diskin communicated in 2007.

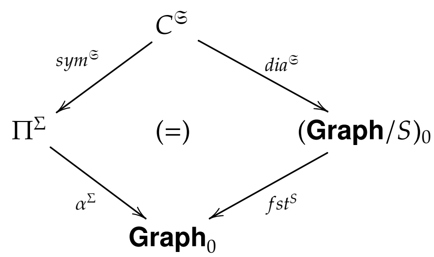
$C^{\mathfrak{S}} \subseteq \Pi^{\Sigma} \times_{\alpha^{\Sigma}} (\mathbf{Graph}/S)_0$.



The important observation is that the maps $sym^{\mathfrak{S}} := in^{\mathfrak{S}}; p_1$ and $dia^{\mathfrak{S}} := in^{\mathfrak{S}}; p_2$ make the square commute and are jointly injective.

We moved from simple graphs to multi graphs by introducing (identifiers for) edges independent of vertices and by dropping the requirement that an edge is uniquely determined by its source and target. Analogously, we introduce now (identifiers for) constraints independent of predicate symbols and carrier graphs and we drop the requirement that $sym^{\mathfrak{S}}$ and $dia^{\mathfrak{S}}$ are jointly monic.

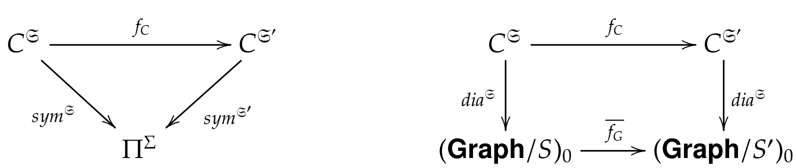
Definition 5.1 (Generalized specification)**.** *Given a signature $\Sigma = (\Pi^{\Sigma}, \alpha^{\Sigma})$, a generalized specification $\mathfrak{S} = (S, C^{\mathfrak{S}}, sym^{\mathfrak{S}}, dia^{\mathfrak{S}})$ consists of a multi graph S, a set $C^{\mathfrak{S}}$ of "constraint identifier's" and two maps $sym^{\mathfrak{S}} : C^{\mathfrak{S}} \to \Pi^{\Sigma}$ and $dia^{\mathfrak{S}} : C^{\mathfrak{S}} \to \mathbf{Graph}_0$ such that the following diagram commutes*



Remark 5.1 (Generalized atomic constraints)**.** *In practice, the "constraint identifiers" will be often constructed as triples (l, π, δ) with l a "label/tag" (indicating, for example, whom introduced the constraint), π a predicate symbol and $\delta : \alpha^{\Sigma}(\pi) \to S$ a graph homomorphism. In those practical cases, the maps $sym^{\mathfrak{S}}$ and $dia^{\mathfrak{S}}$ are given by the second and third projection, respectively.*

For the definition of morphisms we have to remind that any graph homomorphism $\phi_G : S \to S'$ induces a functor $\overline{\phi_G} : (\mathbf{Graph}/S) \to (\mathbf{Graph}/S')$ with $\overline{\phi_G}; fst^{S'} = fst^S$ which is defined by simple post-composition, i.e., $\overline{\phi_G}(\gamma) := \gamma; \phi_G$ for all objects $\gamma : G \to S$ in $\mathbf{Graph}/S$.

Definition 5.2 (Morphisms between generalized specifications)**.** *Given two generalized specifications $\mathfrak{S} = (S, C^{\mathfrak{S}}, sym^{\mathfrak{S}}, dia^{\mathfrak{S}})$ and $\mathfrak{S}' = (S', C^{\mathfrak{S}'}, sym^{\mathfrak{S}'}, dia^{\mathfrak{S}'})$, a specification morphism $f = (f_C, f_G) : \mathfrak{S} \to \mathfrak{S}'$ is given by a mapping $f_C : C^{\mathfrak{S}} \to C^{\mathfrak{S}'}$ and a graph homomorphism $f_G : S \to S'$ such that the following two diagrams commute:*



Definition 5.3 (Category of generalized specifications)**.** *Given a signature $\Sigma = (\Pi^{\Sigma}, \alpha^{\Sigma})$, the category* $\mathbf{GSpec}(\Sigma)$ *has all generalized specifications $\mathfrak{S} = (S, C^{\mathfrak{S}}, sym^{\mathfrak{S}}, dia^{\mathfrak{S}})$ as objects and all generalized specification morphisms $\phi : \mathfrak{S} \to \mathfrak{S}'$ as morphisms between generalized specifications $\mathfrak{S}$ and $\mathfrak{S}'$.*

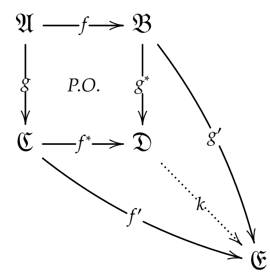
The composition $\phi; \psi : \mathfrak{G} \to \mathfrak{K}$ of two (generalized) specification morphisms $\phi : \mathfrak{G} \to \mathfrak{H}$ and $\psi : \mathfrak{H} \to \mathfrak{K}$ is defined component-wise $\phi; \psi = (\phi_C, \phi_G); (\psi_C, \psi_G) := (\phi_C; \psi_C, \phi_G; \psi_G)$. The identity (generalized) specification morphism $id^G : \mathfrak{G} \to \mathfrak{G}$ is also defined component-wise $id^G = (id^{C^{\mathfrak{G}}}, id^G)$. This ensures that the composition of specification monomorphisms is associative and that identity specification morphism are left and right neutral with respect to composition.

The definitions above give us a category of generalized specifications and it should be possible to prove now that pushouts and pullbacks in this category are given by pushouts respectively pullbacks of the underlying graphs plus the pushout respectively pullbacks of the corresponding sets of identifiers! Since we used in the definitions above category **Graph** as underlying category which is adhesive as well as category **Set** we get as result that the category of generalized specifications **GSpec** is adhesive.

5.1 Pushouts and Pullbacks of Generalized DPF Specifications

In the following we show that a pushout respectively pullback in category **GSpec** are indeed given by the pushout respectively pullback in the underlying category of graphs as wells as the pushout respectively pullbacks of the corresponding sets of identifiers.

Proposition 5.1 (Pushout of generalized specifications). *A pushout* $\mathfrak{B} \xrightarrow{g^*} \mathfrak{D} \xleftarrow{f^*} \mathfrak{C}$ *of a span* $\mathfrak{B} \xleftarrow{f} \mathfrak{A} \xrightarrow{g} \mathfrak{C}$ *of generalized specification morphisms is obtained by constructing a pushout in* **Graph** *for the underlying graph homomorphisms as well as constructing a pushout in* **Set** *for the underlying maps between sets of constraint identifiers.*

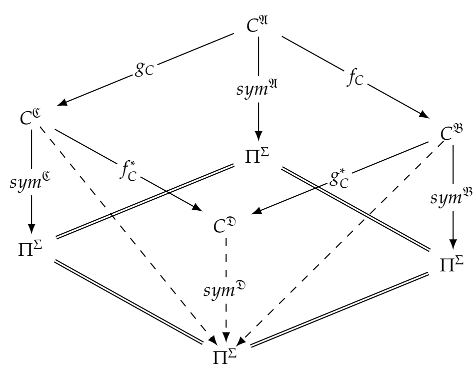


Proof.

"⇐": Component-wise pushouts in Graph and Set provide pushouts in GSpec.

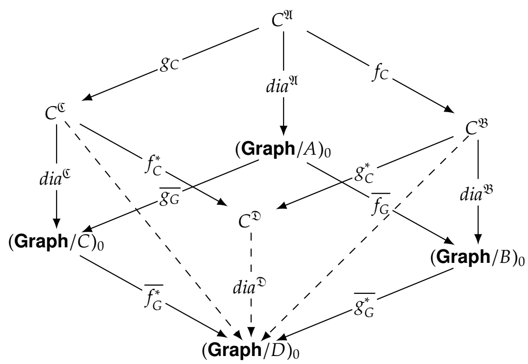
We consider the pushout $C^{\mathfrak{B}} \xrightarrow{g^*_C} C^{\mathfrak{D}} \xleftarrow{f^*_C} C^{\mathfrak{C}}$ of the span $C^{\mathfrak{B}} \xleftarrow{f_C} C^{\mathfrak{A}} \xrightarrow{g_C} C^{\mathfrak{C}}$ in **Set** and the pushout $B \xrightarrow{g^*_G} D \xleftarrow{f^*_G} C$ of the span $B \xleftarrow{f_G} A \xrightarrow{g_G} C$ in **Graph** v.

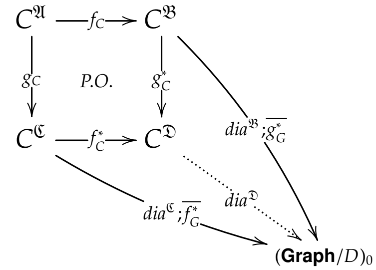
Existence of symbol map:



Since (f_C, f_G) and (g_C, g_G) are specification morphisms we have $g_C; sym^{\mathfrak{C}} = sym^{\mathfrak{A}} = f_C; sym^{\mathfrak{B}}$ thus there exists a unique map $sym^{\mathfrak{D}} : C^{\mathfrak{D}} \to \Pi^{\Sigma}$ with $g^*_C; sym^{\mathfrak{D}} = sym^{\mathfrak{B}}$ and $f^*_C; sym^{\mathfrak{D}} = sym^{\mathfrak{C}}$.

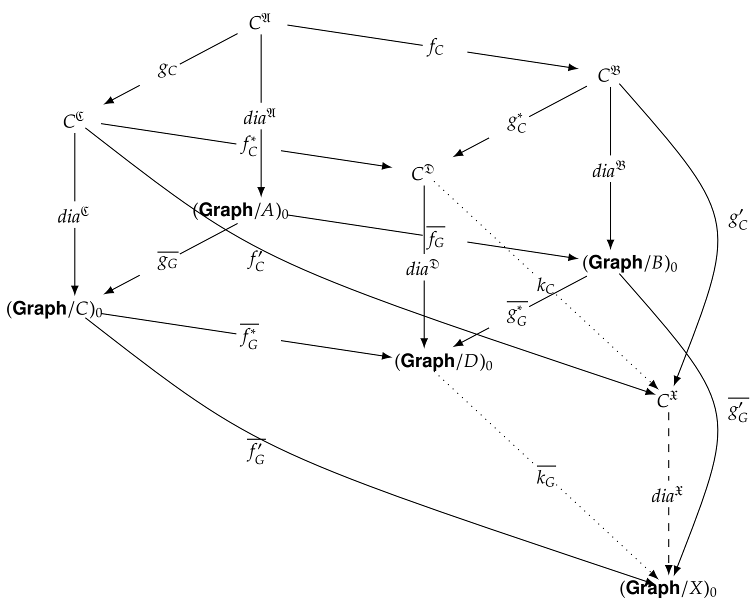
Existence of diagram map: The top face of the cube in the figure below shows the pushout for constraints in **Set** while the commutative square induced by the pushout of the underlying graph homomorphisms is shown in its bottom face. Analogously to Proposition 3.1 we obtain $g_C; dia^{\mathfrak{C}}; \overline{f^*_G} = f_C; dia^{\mathfrak{B}}; \overline{g^*_G}$ thus there exists a unique map $dia^{\mathfrak{D}} : C^{\mathfrak{D}} \to (\mathbf{Graph}/S)_0$ with $g^*_C; dia^{\mathfrak{D}} = dia^{\mathfrak{B}}; \overline{g^*_G}$ and $f^*_C; dia^{\mathfrak{D}} = dia^{\mathfrak{C}}; \overline{f^*_G}$.

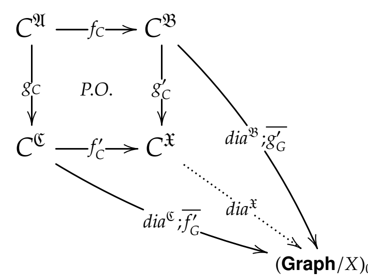




Morphism property: The equations above ensure that the pair (g^*_C, g^*_G) defines a specification morphism $g^* : \mathfrak{B} \to \mathfrak{D}$ and that the pair (f^*_C, f^*_G) defines a specification morphism $f^* : \mathfrak{C} \to \mathfrak{D}$, respectively.

Universal property: Assume specification $\mathfrak{X}$ and two specification morphisms $f' : \mathfrak{C} \to \mathfrak{X}$ and $g' : \mathfrak{B} \to \mathfrak{X}$ so that $g; f' = f; g'$ (see figures below).



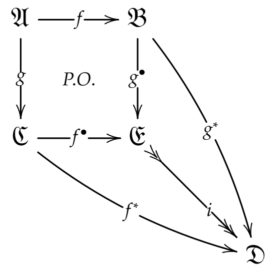


1. Due to the pushout in **Set**, there is a unique mediating mapping k_C with $g^*_C;k_C = g'_C$, $f^*_C;k_C = f'_C$ and, due to the pushout in **Graph**, there is a unique graph homomorphism k_G with $g^*_G;k_G = g'_G$, $f^*_G;k_G = f'_G$. This ensures, especially, that the bottom face in the diagram above is commutative.

2. By the construction of (k_C, k_G) and the morphism property of f', f^*, g' and g^*, respectively, we get $f^*_C;k_C;dia^{\mathfrak{X}} = f^*_C;dia^{\mathfrak{D}};\overline{k_G}$, and $g^*_C;k_C;dia^{\mathfrak{X}} = g^*_C;dia^{\mathfrak{D}};\overline{k_G}$ thus uniqueness of mediating morphisms for the top pushout entails $k_C;dia^{\mathfrak{X}} = dia^{\mathfrak{D}};\overline{k_G}$. This means that the unique pair (k_C, k_G) defines indeed a specification morphism $k : \mathfrak{D} \to \mathfrak{X}$.

"⇒": Pushouts in GSpec imply pushouts of its components in Graph and Set

Assume a pushout $\mathfrak{B} \xrightarrow{g^\bullet} \mathfrak{E} \xleftarrow{f^\bullet} \mathfrak{C}$ of a span $\mathfrak{B} \xleftarrow{f} \mathfrak{A} \xrightarrow{g} \mathfrak{C}$ of generalized specification morphisms:



As shown we can also construct componentwise a pushout $\mathfrak{B} \xrightarrow{g^*} \mathfrak{D} \xleftarrow{f^*} \mathfrak{C}$ of the same span.

$\Longrightarrow$ Since pushouts are unique up to isomorphism we get an isomorphism $i = (i_C, i_G) : \mathfrak{D} \to \mathfrak{E}$ in **GSpec**.

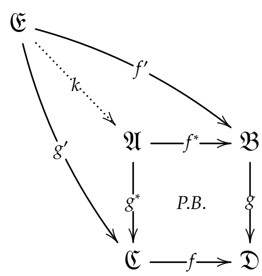
$\Longrightarrow$ Since composition and identities in **GSpec** are given by composition and identities in **Set** and **Graph**, respectively, i_C and i_G become isomorphisms in **Set** and **Graph**, respectively.

$\Longrightarrow$ Since pushouts are closed under isomorphisms this means that the components of $\mathfrak{B} \xrightarrow{g^\bullet} \mathfrak{E} \xleftarrow{f^\bullet} \mathfrak{C}$ in **Set** and **Graph** also constitute pushouts in **Set** and **Graph**, respectively.

□

We have seen that a pushout in **GSpec** can be obtained by constructing a pushout in **Graph** for the underlying graph homomorphisms as well as constructing a pushout in **Set** for the underlying mappings between sets of constraint identifiers. Now, we show that also pullbacks can be obtained in an analogue manner.

Proposition 5.2 (Pullback of generalized specifications)**.** *A pullback $\mathfrak{B} \xleftarrow{f^*} \mathfrak{A} \xrightarrow{g^*} \mathfrak{C}$ of a co-span $\mathfrak{B} \xrightarrow{g} \mathfrak{D} \xleftarrow{f} \mathfrak{C}$ in* **GSpec** *is obtained by constructing a pullback in* **Graph** *for the underlying graph homomorphisms and a pullback in* **Set** *for the underlying mappings between sets of constraint identifiers.*



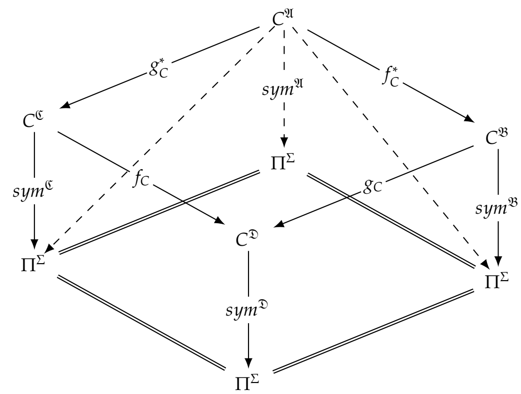
Lemma 5.1. *For any pullback $B \xleftarrow{f^*_G} A \xrightarrow{g^*_G} C$ of a co-span $B \xrightarrow{g_G} D \xleftarrow{f_G} C$ in* **Graph** *the span $(\textbf{Graph}/B)_0 \xleftarrow{\overline{f^*_G}} (\textbf{Graph}/A)_0 \xrightarrow{\overline{g^*_G}} (\textbf{Graph}/C)_0$ is a pullback in* **Set** *of the co-span $(\textbf{Graph}/B)_0 \xrightarrow{\overline{g_G}} (\textbf{Graph}/D)_0 \xleftarrow{\overline{f_G}} (\textbf{Graph}/C)_0$.*

Proof.

"⇐": Componentwise pullbacks in Graph and Set provide pullbacks in GSpec

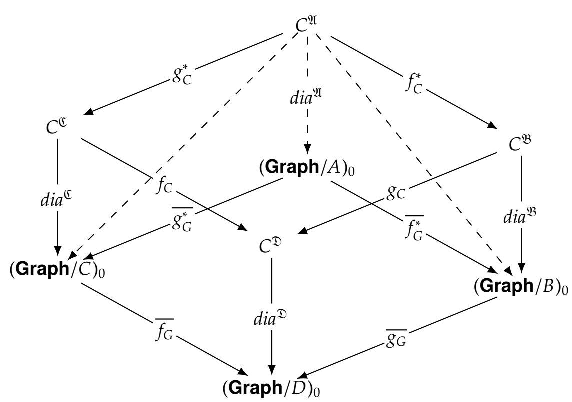
We consider the pullback $C^{\mathfrak{B}} \xleftarrow{f^*_C} C^{\mathfrak{A}} \xrightarrow{g^*_C} C^{\mathfrak{C}}$ of the co-span $C^{\mathfrak{B}} \xrightarrow{g_C} C^{\mathfrak{D}} \xleftarrow{f_C} C^{\mathfrak{C}}$ in **Set** and the pullback $B \xleftarrow{f^*_G} A \xrightarrow{g^*_G} C$ of the co-span $B \xrightarrow{g_G} D \xleftarrow{f_G} C$ in **Graph**.

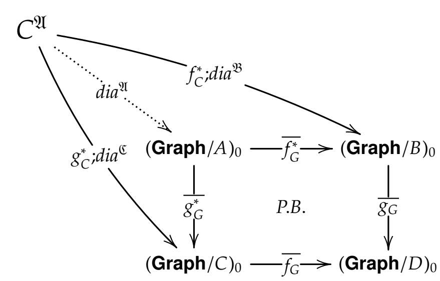
Existence of symbol map:



Since (f_C, f_G) and (g_C, g_G) are specification morphisms we have $f_C^*; sym^{\mathfrak{B}} = g_C^*; sym^{\mathfrak{C}}$ thus there exists a unique map $sym^{\mathfrak{A}} : C^{\mathfrak{A}} \to \Pi^{\Sigma}$ with $f_C^*; sym^{\mathfrak{B}} = sym^{\mathfrak{A}}$ and $g_C^*; sym^{\mathfrak{C}} = sym^{\mathfrak{A}}$.

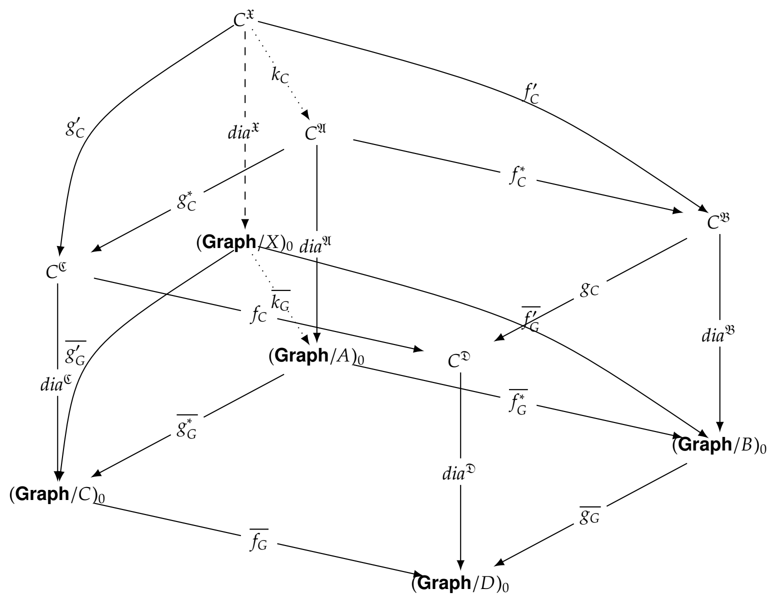
Existence of diagram map: The top face of the cube in the figure below shows the pullback for constraints in **Set** while the pullback according to Lemma 5.1 is shown in the bottom face. Analogously to Proposition 3.2 we obtain $g_C^*; dia^{\mathfrak{C}}; \overline{f_G} = f_C^*; dia^{\mathfrak{B}}; \overline{g_G}$ thus there exists a unique map $dia^{\mathfrak{A}} : C^{\mathfrak{A}} \to (\mathbf{Graph}/S)_0$ with $dia^{\mathfrak{A}}; \overline{f_G^*} = f_C^*; dia^{\mathfrak{B}}$ and $dia^{\mathfrak{A}}; \overline{g_G^*} = g_C^*; dia^{\mathfrak{C}}$.

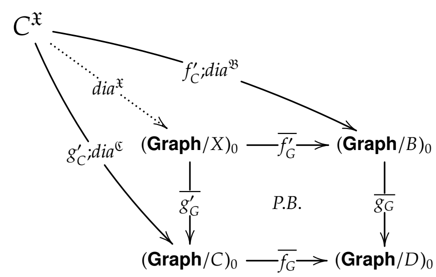




Morphism property: The equations above ensure that the pair (f_C^*, f_G^*) defines a specification morphism $f^* : \mathfrak{A} \to \mathfrak{B}$ and that the pair (g_C^*, g_G^*) defines a specification morphism $g^* : \mathfrak{A} \to \mathfrak{C}$, respectively.

Universal property: Assume a generalized specification $\mathfrak{X}$ and two generalized specification morphisms $f' : \mathfrak{X} \to \mathfrak{B}$ and $g' : \mathfrak{X} \to \mathfrak{C}$ so that $g'; f = f'; g$ (see figures below).



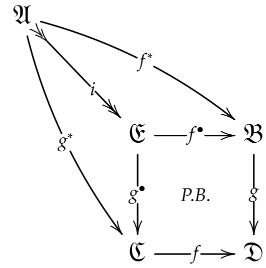


1. Due to the pullback in **Set**, there is a unique mediating mapping k_C with $k_C; f^*_C = f'_C$, $k_C; g^*_C = g'_C$ and, due to the pullback in **Graph**, there is a unique graph homomorphism k_G with $k_G; f^*_G = f'_G$, $k_G; g^*_G = g'_G$. This ensures, especially, that the bottom face in the diagram above is commutative.

2. By the construction of (k_C, k_G) and the morphism property of f', f^*, g' and g^*, respectively, we get $k_C; dia^{\mathfrak{A}}; \overline{f^*_G} = dia^{\mathfrak{X}}; \overline{k_G}; \overline{f^*_G}$, and $k_C; dia^{\mathfrak{A}}; \overline{g^*_G} = dia^{\mathfrak{X}}; \overline{k_G}; \overline{g^*_G}$, thus uniqueness of mediating morphisms for the bottom pullback entails $k_C; dia^{\mathfrak{A}} = dia^{\mathfrak{X}}; \overline{k_G}$. This means that the unique pair (k_C, k_G) defines indeed a specification morphism $k : \mathfrak{X} \to \mathfrak{A}$.

"⇒": Pullbacks in **GSpec** imply pullbacks of its components in **Graph** and **Set**

Assume a pullback $\mathfrak{B} \xleftarrow{f^\bullet} \mathfrak{E} \xrightarrow{g^\bullet} \mathfrak{C}$ of a cospan $\mathfrak{B} \xrightarrow{g} \mathfrak{D} \xleftarrow{f} \mathfrak{C}$ of generalized specification morphisms:



As shown we can also construct componentwise a pullback $\mathfrak{B} \xleftarrow{f^*} \mathfrak{A} \xrightarrow{g^*} \mathfrak{C}$ of the same co-span.

$\Longrightarrow$ Since pullbacks are unique up to isomorphism we get an isomorphism $i = (i_C, i_G) : \mathfrak{A} \to \mathfrak{E}$ in **GSpec**.

$\Longrightarrow$ Since composition and identities in **GSpec** are given by composition and identities in **Set** and **Graph**, respectively, i_C and i_G become isomorphisms in **Set** and **Graph**, respectively.

$\Longrightarrow$ Since pullbacks are closed under isomorphisms this means that the components of $\mathfrak{B} \xleftarrow{f^\bullet} \mathfrak{E} \xrightarrow{g^\bullet} \mathfrak{C}$ in **Set** and **Graph** also constitute pushouts in **Set** and **Graph**, respectively.

□

5.2 Van Kampen Property for Generalized DPF Specifications

In the following, we consider the VK property for **GSpec**. Since, this time the construction of pushouts does not rely on a union construction anymore, we expect the category to be adhesive.

Corollary 5.1 (Monomorphism in (generalized) specifications)**.** *Monomorphisms in* **GSpec** *are given by componentwise monomorphisms in* **Set** *respectively* **Graph** *i.e. a specification morphism $f = (f_C, f_G) : \mathfrak{A} \to \mathfrak{B}$ is a monomorphism iff $f_C : C^{\mathfrak{A}} \to C^{\mathfrak{B}}$ is a monomorphism in* **Set** *and $f_G : A \to B$ is a monomorphism in* **Graph***.*

Proof. Since the identities in **GSpec** are componentwise identities we get by Lemma 3.1, Proposition 5.2 and again Lemma 3.1 for any generalized specification morphism in $f : \mathfrak{A} \to \mathfrak{B}$:

f is a monomorphism in **GSpec**

$\Longleftrightarrow$ $\mathfrak{A} \xleftarrow{id^A} \mathfrak{A} \xrightarrow{id^A} \mathfrak{A}$ is the pullback of $\mathfrak{A} \xrightarrow{f} \mathfrak{B} \xleftarrow{f} \mathfrak{A}$

$\Longleftrightarrow$ $A \xleftarrow{id^A} A \xrightarrow{id^A} A$ is the pullback of $A \xrightarrow{f_G} B \xleftarrow{f_G} A$ in **Graph** and $C^{\mathfrak{A}} \xleftarrow{id^{C^{\mathfrak{A}}}} C^{\mathfrak{A}} \xrightarrow{id^{C^{\mathfrak{A}}}} C^{\mathfrak{A}}$ is the pullback of $C^{\mathfrak{A}} \xrightarrow{f_C} C^{\mathfrak{B}} \xleftarrow{f_C} C^{\mathfrak{A}}$ in **Set**

$\Longleftrightarrow$ $f_G : A \to B$ is a monomorphism in **Graph** and $f_C : C^{\mathfrak{A}} \to C^{\mathfrak{B}}$ is a monomorphism in **Set**

□

Corollary 5.2 (Epimorphism in (generalized) specifications)**.** *Epimorphisms in* **GSpec** *are given by componentwise monomorphisms in* **Set** *respectively* **Graph** *i.e. a specification morphism $f = (f_C, f_G) : \mathfrak{A} \to \mathfrak{B}$ is an epimorphism iff $f_C : C^{\mathfrak{A}} \to C^{\mathfrak{B}}$ in* **Set** *and $f_G : A \to B$ is an epimorphism in* **Graph***.*

Proof. Since the identities in **GSpec** are componentwise identities we get by Lemma 3.2, Proposition 5.1 and again Lemma 3.2 for any generalized specification morphism in $f : \mathfrak{A} \to \mathfrak{B}$:

f is an epimorphism in **GSpec**

$\Longleftrightarrow$ $\mathfrak{B} \xrightarrow{id^B} \mathfrak{B} \xleftarrow{id^B} \mathfrak{B}$ is the pushout of $\mathfrak{B} \xleftarrow{f} \mathfrak{A} \xrightarrow{f} \mathfrak{B}$

$\Longleftrightarrow$ $B \xrightarrow{id^B} B \xleftarrow{id^B} B$ is the pushout of $B \xleftarrow{f_G} A \xrightarrow{f_G} B$ in **Graph** and $C^{\mathfrak{B}} \xrightarrow{id^{C^{\mathfrak{B}}}} C^{\mathfrak{B}} \xleftarrow{id^{C^{\mathfrak{B}}}} C^{\mathfrak{B}}$ is the pushout of $C^{\mathfrak{B}} \xleftarrow{f_C} C^{\mathfrak{A}} \xrightarrow{f_C} C^{\mathfrak{B}}$ in **Set**

$\Longleftrightarrow$ $f_G : A \to B$ is an epimorphism in **Graph** and $f_C : C^{\mathfrak{A}} \to C^{\mathfrak{B}}$ is an epimorphism in **Set**

□

Considering all facts about the category of generalize DPF specification **GSpec**, we can conclude that it is adhesive [3] and that it inherits adhesiveness from category **Set**.

Proposition 5.3. *Pushouts along monomorphism are VK squares in* **GSpec***.*

Proof. Assume a commutative cube (2) in **GSpec** (See Definition 1.1). Since composition of generalized specification morphisms is defined componentwise we have (2) commutes in **GSpec** iff the corresponding two cubes for constraints commute in **Set** and for graphs commute in **Graph**[3]. We assume top face in (2) is a pushout and both back faces in (2) are pullbacks. By Proposition 5.1 and Proposition 5.2 follows:

[3] That implies that the corresponding cubes for vertices and edges commute in **Set**.

top face in (1) is a pushout	$\iff$	top face in (1) for (1_C) and (1_G) are pushouts and
back faces in (2) are pullbacks	$\iff$	back faces in (2) for (2_C) and (2_G) are pullbacks

We have by Proposition 3.1, Proposition 3.2 and **Set** is adhesive:

bottom face in (2) is pushout	$\iff$	bottom faces in (2_C) and (2_G) are pushouts
	$\iff$	front faces in (2_C) and (2_G) are pullbacks
	$\iff$	front faces in (2) are pullbacks

□

Let us review Figure 4 which shows a counterexample for the VK property in **Spec**. The cube in the figure is not a valid VK cube even top and bottom faces are pushouts and both back faces are pullbacks. Figure 6 shows the analog example for **GSpec**. This time we also have to map the constraints since we can have more than one constraint with the same predicate symbol and the same arity mapping. In contrast to the earlier example we get two constraints in the upper pushout graph. Therefore, we can correctly trace back the constraints in the right front face of the cube which shows in contrast to Figure 4 a valid pullback diagram. Hence Figure 6 shows a valid VK cube.

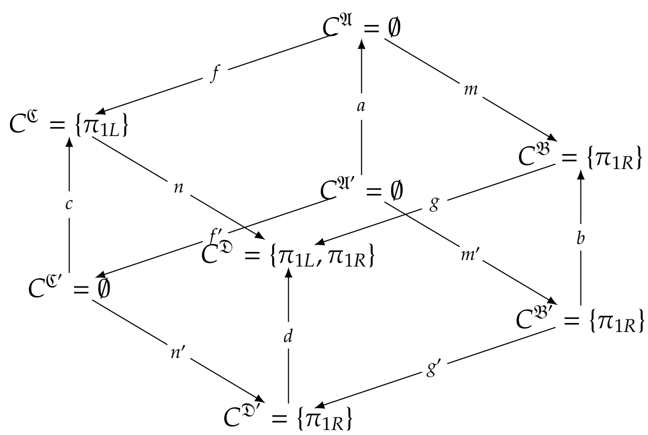


Figure 6: Example Van Kampen property in **GSpec**

6 Conclusion

In this report we have examined how DPF's category of specifications fits into the context of adhesive categories. First, we showed that the concept of DPF specification, as it has been defined and used up to now, does not provide adhesive categories of specifications. Then we presented a generalization of the concept of specification and showed that this generalization provides us with adhesive categories **GSpec**. Hence, results known from graph transformations can be used with DPF and vice versa. Furthermore, we showed that this generalization step is analogous to the step going from the category **SGraph** of simple directed graphs to the category **Graph** of directed multi graphs.

In this paper the definition of both categories **Spec** and **GSpec** is based on the category **Graph**. Instead of **Graph** we could however use other base categories **Base** [2] to define corresponding categories **Spec** and **GSpec**. There shouldn't be any problems to generalize the concepts and results, presented in this paper, to arbitrary adhesive base categories **Base**. Also in this general case the step from category **Spec** to category **GSpec** has the effect that **GSpec** is adhesive while **Spec** is not. The same result we should get for adhesive HLR categories.

However, in [3] also weak adhesive HLR categories are considered as a suitable framework for algebraic graph transformations. In contrast to adhesive HLR categories, weak adhesive categories have stricter conditions for Van Kampen squares. We guess that choosing a weak adhesive category as base category **Base** has the effect that **GSpec** is also weak adhesive. Another interesting research question is under which conditions already category **Spec** with a weak adhesive HLR category as base category is weak adhesive.

References

[1] Z. Diskin. Generalized sketches as an algebraic graph-based framework for semantic modeling and database design. Technical Report 9701, University of Latvia, Riga, Latvia, August 1997. (document)

[2] Z. Diskin and U. Wolter. A Diagrammatic Logic for Object-Oriented Visual Modeling. In *Proceedings of ACCAT 2007: 2nd Workshop on Applied and Computational Category Theory*, volume 203/6 of *Electronic Notes in Theoretical Computer Science*, pages 19–41. Elsevier, 2008. doi: 10.1016/j.entcs.2008.10.041.

[3] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. Springer, March 2006. ISBN 978-3-540-31187-4. doi: 10.1007/3-540-31188-2. 1, 2.2, 3, 3.2, 5.2, 6

[4] S. Lack and P. Sobocinski. Adhesive Categories. In I. Walukiewicz, editor, *Proceedings of FoSSaCS 2004: 7th Foundations of Software Science and Computation Structures*, volume 2987 of *Lecture Notes in Computer Science*, pages 273–288, 2004. ISBN 3-540-21298-1. doi: 10.1007/978-3-540-24727-2_20. 1

[5] M. Makkai. Generalized Sketches as a Framework for Completeness Theorems. *Journal of Pure and Applied Algebra*, 115(1):49–79, 179–212, 214–274, 1997. doi: 10.1016/S0022-4049(96)00007-2. (document)

[6] A. Rossini. *Diagram Predicate Framework meets Model Versioning and Deep Metamodelling*. PhD thesis, Department of Informatics, University of Bergen, Norway, 2011. 4, 4.1, 1

[7] A. Rutle. *Diagram Predicate Framework: A Formal Approach to MDE*. PhD thesis, Department of Informatics, University of Bergen, Norway, 2010. 4, 4.1, 1

[8] U. Wolter and Z. Diskin. From Indexed to Fibred Semantics – The Generalized Sketch File. Technical Report 361, Department of Informatics, University of Bergen, Norway, October 2007. 4.1