Technische Universität Berlin

Forschungsberichte
der Fakultät IV – Elektrotechnik und Informatik

Generalized Typed Attributed Graph Transformation Systems based on Morphisms Changing Type Graphs and Data Signature

Hartmut Ehrig, Karsten Ehrig,
Claudia Ermel, and Ulrike Prange

Bericht-Nr. 2009-08
ISSN 1436-9915

Generalized Typed Attributed Graph Transformation Systems based on Morphisms Changing Type Graphs and Data Signatures

Hartmut Ehrig[1], Karsten Ehrig[2], Claudia Ermel[1], and Ulrike Prange[1]

[1] Technische Universität Berlin, Germany
ehrig|lieske|uprange@cs.tu-berlin.de
[2] University of Leicester, United Kingdom
karsten@mcs.le.ac.uk

Abstract. Our aim is to extend the framework of typed attributed graphs in [1] to generalized typed attributed graphs. They are based on generalized attributed graph morphisms, short GAG-morphisms, which allow to change the type graph, data signature, and domain. This allows to formulate type hierarchies and views of visual languages defined by GAG-morphisms between type graphs, short GATG-morphisms. In order to study

- interaction and integration of views,
- restriction of views along type hierarchies,
- restriction and integration of consistent view models and
- reflection of behaviour between different typed attributed graph transformation systems

we present suitable conditions for the construction of pushouts and pullbacks, and special van Kampen properties in the category **GAGraphs** of generalized attributed graphs. Moreover, we show that $(\mathbf{GAGraphs}, \mathcal{M})$ and $(\mathbf{GAGraphs_{ATG}}, \mathcal{M})$ are adhesive HLR categories for the class $\mathcal{M}$ of injective, persistent, and signature preserving morphisms.

1 Generalized Attributed Graph Morphisms

According to [1] attributed graphs are defined by

Definition 1 (Attributed graph). *An* attributed graph $AG = (G, DSIG, D)$ *consists of*

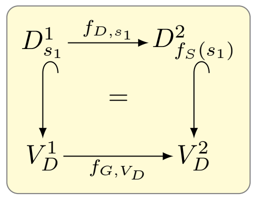
- *an E-graph* $G = (V_G, V_D, E_G, E_{NA}, E_{EA}, (source_j, target_j)_{j \in \{G,NA,EA\}})$,
- *a data signature* $DSIG = (S, S_D, OP)$ *with attribute value sorts* $S_D \subseteq S$, *and*
- *a DSIG-algebra D such that* $\dot{\bigcup}_{s \in S_D} D_s = V_D$.

In addition to attributed graph morphisms as presented in [1], generalized attributed graph morphisms are mappings of attributed graphs with possibly different data signatures.

Definition 2 (Generalized attributed graph morphism). *Given attributed graphs* $AG^i = (G^i, DSIG^i, D^i)$ *for* $i = 1, 2$*, a* generalized attributed graph morphism *(GAG-morphism)* $f = (f_G, f_S, f_D) : AG^1 \to AG^2$ *is given by*

- *an E-graph morphism* $f_G : G^1 \to G^2$*,*
- *a signature morphism* $f_S : DSIG^1 \to DSIG^2$*, and*
- *a generalized homomorphism* $f_D : D^1 \to D^2$*, which is a* $DSIG^1$*-morphism* $f_D : D^1 \to V_{f_S}(D^2)$ *with* $f_D = (f_{D,s_1} : D^1_{s_1} \to D^2_{f_S(s_1)})_{s_1 \in S^1}$

with the following compatibility property*:* $f_S(S^1_D) \subseteq S^2_D$ *and the following diagram commutes for all* $s_1 \in S^1_D$*.*



Definition 3 (Category GAGraphs). *Attributed graphs with generalized attributed graph morphisms and the usual definition of composition and identity form the category* **GAGraphs***.*

According to [1], attributed type graphs and typed attributed graphs are defined by

Definition 4 (Attributed type graph). *An attributed type graph* $ATG = (TG, DSIG, Z_{DSIG})$ *is an attributed graph, where* Z_{DSIG} *is the final DSIG-algebra, i.e.* $Z_{DSIG,s} = \{s\}$ *for all* $s \in S$*, and* $V_D = \dot{\bigcup}_{s \in S_D} Z_{DSIG,s} = S_D$*.*

Definition 5 (Typed attributed graph). *Given an attributed type graph* ATG*, a typed attributed graph* $TAG = (AG, t)$ *(over* ATG*) is given by an attributed graph* AG *and a GAG-morphism* $t : AG \to ATG$*.*

Definition 6 (Typed attributed graph morphism). *Given an attributed type graph* ATG *and typed attributed graphs* $TAG^i = (AG^i, t : AG^i \to ATG)$ *over* ATG *for* $i = 1, 2$*, a typed attributed graph morphism* $f : TAG^1 \to TAG^2$ *is given by a GAG-morphism* $f : AG^1 \to AG^2$ *such that* $t_2 \circ f = t^1$*.*

Definition 7 (Category $\mathbf{GAGraphs_{ATG}}$). *Given an attributed type graph* ATG*, typed attributed graphs over* ATG *and typed attributed graph morphisms form the category* $\mathbf{GAGraphs_{ATG}}$*.*

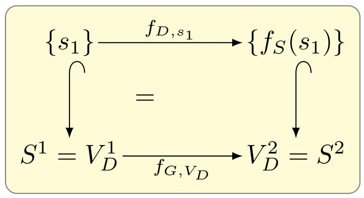
Remark 1. $\mathbf{GAGraphs_{ATG}} \cong \mathbf{GAGraphs}\backslash ATG$ (slice category).

As a special case of Def. 2 and Def. 4 we obtain

Definition 8 (Generalized attributed type graph morphism). *Given attributed type graphs* $ATG^i = (TG^i, DSIG^i, Z_{DSIG^i})$ *for* $i = 1, 2$*, a* generalized attributed type graph morphism *(GATG-morphism)* $f = (f_G, f_S, f_D) : ATG^1 \to ATG^2$ *is given by*

– *an E-graph morphism* $f_G : TG^1 \to TG^2$*,*
– *a signature morphism* $f_S : DSIG^1 \to DSIG^2$*, and*
– *a generalized homomorphism* $f_D : Z_{DSIG^1} \to Z_{DSIG^2}$*, which is uniquely determined by* $f_{D,s_1}(s_1) = f_S(s_1)$ *for all* $s_1 \in S^1$*.*

Remark 2. A generalized attributed type graph morphism f is also a generalized attributed graph morphism since the compatibility property is automatically satisfied. This is shown in the following diagram, where $f_{G,V_D}(s_1) = f_S(s_1)$ for all $s_1 \in S^1_D$ and f_D, f_{G,V_D} are uniquely determined by f_S.



Definition 9 (properties of GAG-morphisms). *A GAG-morphism* $f = (f_G, f_S, f_D) : (G^1, DSIG^1, D^1) \to (G^2, DSIG^2, D^2)$ *is called*

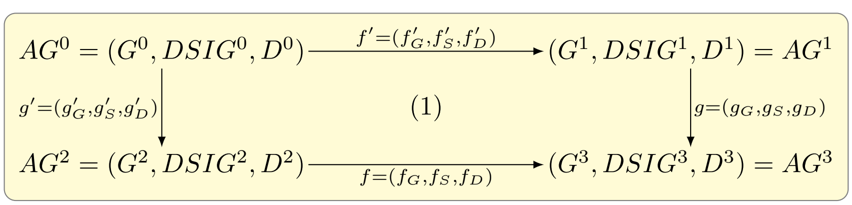
1. injective*, if* f_G*,* f_S*,* f_D *are injective,*
2. signature preserving*, if* f_S *is isomorphic,*
3. persistent*, if* f_D *is isomorphic.*

Remark 3. By definition we have

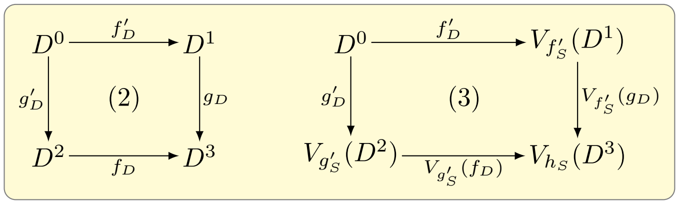
– f AG-morphism $\Leftrightarrow$ f signature preserving GAG-morphism,
– f AG-morphism in $\mathcal{M}$ $\Leftrightarrow$ f injective, persistent, signature preserving GAG-morphism,
– f GATG-morphism $\Rightarrow$ f persistent.

Theorem 1 (Pullback construction in GAGraphs). *Given GAG-morphisms* $f : AG^2 \to AG^3$ *and* $g : AG^1 \to AG^3$ *then the following construction* (1) *is a pullback in* **GAGraphs***. Moreover, the pullback construction preserves injective, signature preserving, and persistent morphisms.*

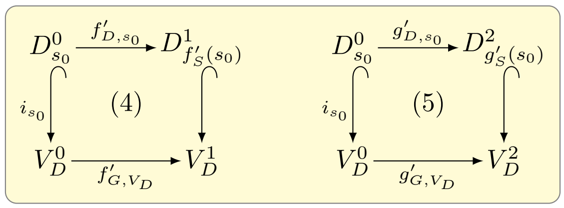
Construction.



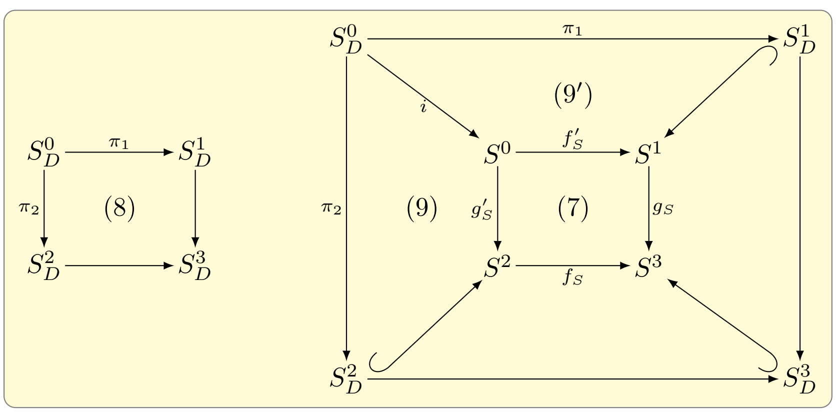
In the G- and S-components, we have pullbacks in the categories **EGraphs** and **Signatures** which are constructed componentwise in **Sets**, with attribute value sorts S^0_D as the corresponding pullback of the attribute value sorts. In the D-component, we have the pullback (2) of generalized algebras given by the pullback (3) in $DSIG^0$-**Algs** which is constructed componentwise in **Sets**, with $h_S = g_S \circ f'_S = f_S \circ g'_S$.



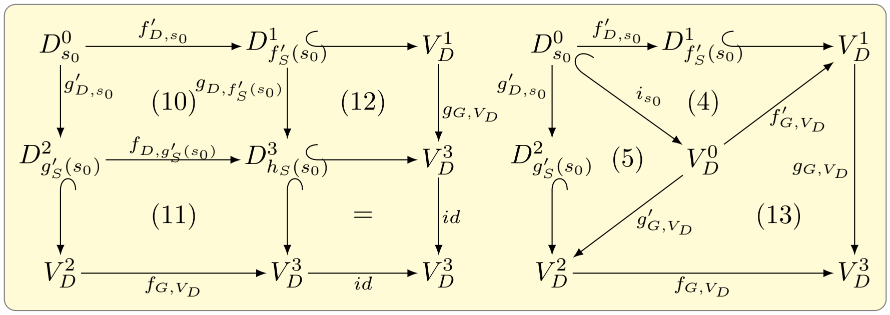
Proof. For the well-definedness of this construction we have to find an injective $i : S_D^0 \to S^0$ and injective $i_{s_0} : D_{s_0}^0 \to V_D^0$ for all $s_0 \in S_D^0$ such that the compatibility diagrams (4) and (5) commute and we have (6) $\dot{\bigcup}_{s_0 \in S_D^0} D_{s_0}^0 = V_D^0$ with coproduct injections i_{s_0}. Finally the universal pullback properties have to be shown.



Consider the pullbacks (7) and (8) as the pullbacks of the sorts and the attribute value sorts in **Sets**, respectively. Since (7) is a pullback and (8) commutes we obtain a unique $i : S_D^0 \to S^0$ such that (9) and (9′) commute. For $x, y \in S_D^0$ with $i(x) = i(y)$ we have that $\pi_1(x) = f'_S(i(x)) = f'_S(i(y)) = \pi_1(y)$ and $\pi_2(x) = g'_S(i(x)) = g'_S(i(y)) = \pi_2(y)$. Since (8) is also a pullback it follows that $x = y$, hence i is injective.



By construction of D^0 as pullback in (2) and (3) we obtain for all $s_0 \in S_D^0$ diagram (10), and (11) and (12) are the compatibility diagrams for GAG-morphisms f and g, respectively. Moreover, let (13) be the V_D-component of the pullback in **EGraphs**, which leads to a unique $i_{s_0} : D_{s_0}^0 \to V_D^0$ such that (4) and (5) commute, where the left diagram shows that the outer diagram on the right commutes.



To show that i_{s_0} is injective, suppose we have $d_1,\ d_2 \in D^0_{s_0}$ with $i_{s_0}(d_1) = i_{s_0}(d_2)$. Then we have that

- (4) commutes and $D^1_{f'_S(s_0)} \to V^1_D$ being injective implies that $f'_{D,s_0}(d_1) = f'_{D,s_0}(d_2)$ and
- (5) commutes and $D^2_{g'_S(s_0)} \to V^2_D$ being injective implies that $g'_{D,s_0}(d_1) = g'_{D,s_0}(d_2)$.

By construction, (10) is a pullback in **Sets** and hence f'_{D,s_0} and g'_{D,s_0} are jointly injective. Thus we have $d_1 = d_2$ and i_{s_0} is injective.

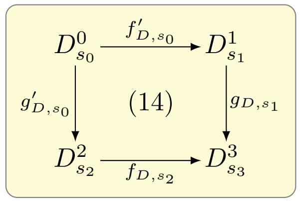
It remains to show (6) or, more precisely, that the injective $i_{s_0} : D^0_{s_0} \to V^0_D$ are coproduct injections, i.e.

1. $i_{s_0}(D^0_{s_0}) \cap i_{s'_0}(D^0_{s'_0}) = \varnothing$ for all $s_0 \neq s'_0 \in S^0_D$ and
2. for all $s_0 \in S^0_D$, i_{s_0} are jointly surjective.

1. Assume $x_0 \in D^0_{s_0}$, $x'_0 \in D^0_{s'_0}$ with $i_{s_0}(x_0) = i_{s'_0}(x'_0) = x \in V^0_D$ with $f'_{G,V_D}(x) = x_1 \in V^1_D$, $g'_{G,V_D}(x) = x_2 \in V^2_D$. Then $x_1 \in D^1_{f'_S(s_0)} \cap D^1_{f'_S(s'_0)}$ by (4) for s_0 and s'_0, and $V^1_D = \dot{\bigcup}_{s_1 \in S^1_D} D^1_{s_1}$ implies $f'_S(s_0) = f'_S(s'_0)$. Analogously, $x_2 \in D^2_{g'_S(s_0)} \cap D^2_{g'_S(s'_0)}$ by (5), and $V^2_D = \dot{\bigcup}_{s_2 \in S^2_D} D^2_{s_2}$ implies $g'_S(s_0) = g'_S(s'_0)$. From the pullback (7) we obtain that f'_S and g'_S are jointly injective, hence it follows that $s_0 = s'_0$, which is a contradiction.
2. Given $x \in V^0_D$ with $f'_{G,V_D}(x) = x_1 \in V^1_D$, $g'_{G,V_D}(x) = x_2 \in V^2_D$, and $x_3 = g_{G,V_D}(x_1) = f_{G,V_D}(x_2) \in V^3_D$. We have to find $s_0 \in S^0_D$ and $x_0 \in D^0_{s_0}$ with $i_{s_0}(x_0) = x$.

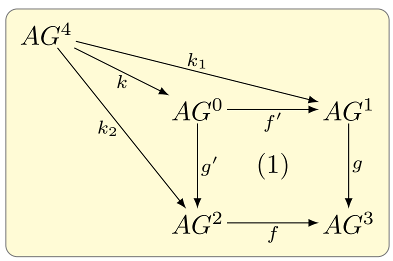
   $x_1 \in V^1_D$ and $V^1_D = \dot{\bigcup}_{s_1 \in S^1_D} D^1_{s_1}$, and $x_2 \in V^2_D$ and $V^2_D = \dot{\bigcup}_{s_2 \in S^2_D} D^2_{s_2}$ imply $\exists! s_1 \in S^1_D$ with $x_1 \in D^1_{s_1}$ and $\exists! s_2 \in S^2_D$ with $x_2 \in D^2_{s_2}$, respectively. By (13) and compatibility of f and g we have that $g_{G,V_D}(x_1) = g_{D,s_1}(x_1) = x_3 = f_{G,V_D}(x_2) = f_{D,s_2}(x_2)$. $x_3 \in V^3_D$ implies that there exists a unique $s_3 \in S^3_D$ with $x_3 \in D^3_{s_3}$. Using $g_{D,s_1}(x_1) = x_3 \in D^3_{s_3}$ implies $s_3 = g_S(s_1)$ by compatibility of g. Similar $f_{D,s_2}(x_2) = x_3 \in D^3_{s_3}$ implies $s_3 = f_S(s_2)$ by compatibility of f.

From the signatur pullback and $g_S(s_1) = s_3 = f_S(s_2)$ we obtain a unique $s_0 \in S_D^0$ with $f'_S(s_0) = s_1$ and $g'_S(s_0) = s_2$. From the data type pullback we obtain the following pullback (14) in **Sets**.

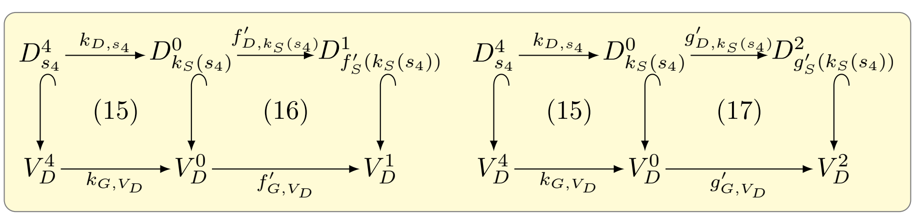


(14) being a pullback and $g_{D,s_1}(x_1) = x_3 = f_{D,s_2}(x_2)$ imply that there exists a unique $x_0 \in D^0_{s_0}$ with $f'_{D,s_0}(x_0) = x_1$ and $g'_{D,s_0}(x_0) = x_2$. Now we have that $f'_{G,V_D} \circ i_{s_0}(x_0) = f'_{D,s_0}(x_0) = x_1$ by (4) and $g'_{G,V_D} \circ i_{s_0}(x_0) = g'_{D,s_0}(x_0) = x_2$ by (5). By construction we have $f'_{G,V_D}(x) = x_1$ and $g'_{G,V_D}(x) = x_2$. Since (13) is a pullback it follows that $i_{s_0}(x_0) = x$ as required.

It remains to show the universal pullback property. The induced morphism $k = (k_G, k_S, k_D)$ is unique in each component by pullback construction in each component, and it suffices to show the compatibility property for $k : AG^4 \to AG^0$. We have to show the commutativity of (15) for all $s_4 \in S_D^4$.



(15+16), (15+17), (16) and (17) commute by compatibility of k_1, k_2, f' and g', respectively. Hence (15) is equalized by f'_{G,V_D} and g'_{G,V_D}, which are jointly monomorphisms from the pullback in the V_D-component. This implies that (15) commutes.



Moreover, the pullback constructions preserve injectivity and isomorphisms of all the different components. This implies that injective, signature preserving, and persistent GAG-morphisms are preserved.

Remark 4. Given a commutative diagram (1) in **GAGraphs** as in the construction with pullbacks in the G-, S-, and D-components then (1) is a pullback in **GAGraphs**. This is a consequence of the fact that the universal pullback property in **GAGraphs** above only requires pullbacks in each component.

Example 1. Given a GATG-morphism $f : ATG^1 \to ATG^2$ and a signature preserving $g : AG^2 \to ATG^2$, by definition of GATG-morphisms this means that f is persistent, which implies that also f' is persistent and g' is signature preserving in the following pullback (1), where G^1 is the pullback of G^2 and TG^1 along TG^2 in **EGraphs**, and persistency of f' implies $D^1_{s_1} = D^2_{f'_S(s_1)}$ for all $s_1 \in S^1$, and hence $D^1 \cong V_{f'_S}(D^2)$.

$$\begin{array}{ccc} AG^1 = (G^1, DSIG^1, D^1) & \xrightarrow{f'} & (G^2, DSIG^2, D^2) = AG^2 \\ \downarrow g' & (1) & \downarrow g \\ ATG^1 = (TG^1, DSIG^1, Z_{DSIG^1}) & \xrightarrow[f]{} & (TG^2, DSIG^2, Z_{DSIG^2}) = ATG^2 \end{array}$$

Definition 10 (Forward and backward typing). *Given a GATG-morphism $f : ATG^1 \to ATG^2$, then we have that*

- *the* forward typing $f^> : \mathbf{GAGraphs_{ATG^1}} \to \mathbf{GAGraphs_{ATG^2}}$ *is given by* $f^>(AG^1 \xrightarrow{t^1} ATG^1) = (AG^1 \xrightarrow{t^1} ATG^1 \xrightarrow{f} ATG^2)$,
- *the* backward typing $f^< : \mathbf{GAGraphs_{ATG^2}} \to \mathbf{GAGraphs_{ATG^1}}$ *is given by* $f^<(AG^2 \xrightarrow{t^2} ATG^2) = (AG^1 \xrightarrow{t^1} ATG^1)$, *where t^1 is given by the following pullback* (1) *in* **GAGraphs**.

$$\begin{array}{ccc} AG^1 & \xrightarrow{f'} & AG^2 \\ \downarrow t^1 & (1) & \downarrow t^2 \\ ATG^1 & \xrightarrow[f]{} & ATG^2 \end{array}$$

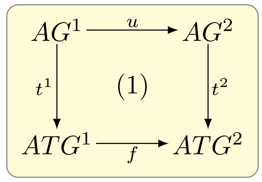
Remark 5. Note that t^1 is signature preserving if this holds for t^2, which allows to restrict $f^<$ to $f^< : \mathbf{AGraphs_{ATG^2}} \to \mathbf{AGraphs_{ATG^1}}$. This restriction does not hold for $f^>$.

Theorem 2 (Forward and backward typing are adjoint). *Given a GATG-morphism $f : ATG^1 \to ATG^2$ then forward typing $f^>$ is left adjoint to backward typing $f^<$*

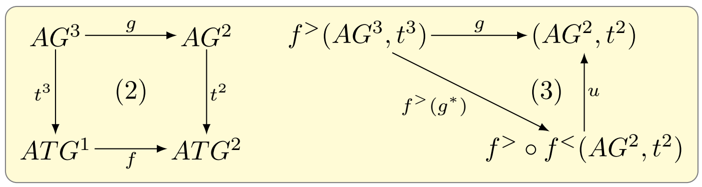
$$f^> \dashv f^< : \mathbf{GAGraphs_{ATG^2}} \to \mathbf{GAGraphs_{ATG^1}}.$$

Proof. Forward typing is a functor $f^> : \mathbf{GAGraphs_{ATG^1}} \to \mathbf{GAGraphs_{ATG^2}}$ defined on morphisms $h : (AG^1_1, t^1_1) \to (AG^1_2, t^1_2)$ by $f^>(h) = h : (AG^1_1, f \circ t^1_1) \to (AG^1_2, f \circ t^1_2)$.

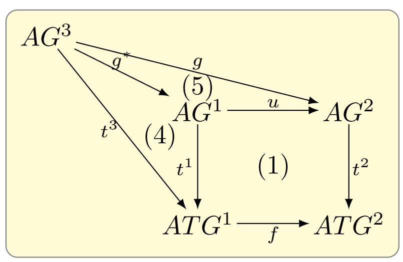
For each (AG^2, t^2) in $\mathbf{GAGraphs_{ATG_2}}$, we define the cofree construction $f^<(AG^2, t^2) = (AG^1, t^1)$ by backward typing in pullback (1) in **GAGraphs**, where the universal morphism $u : f^> \circ f^<(AG^2, t^2) = (AG^1, f \circ t^1) \to (AG^2, t^2)$ in $\mathbf{AGraphs_{ATG^2}}$ is given by $u : AG^1 \to AG^2$.



In order to show the universal property of the cofree construction let $g : f^>(AG^3, t^3) = (AG^3, f \circ t^3) \to (AG^2, t^2)$ in $\mathbf{GAGraphs_{ATG^2}}$ given by (2).



We have to construct a unique $g^* : (AG^3, t^3) \to f^<(AG^2, t^2) = (AG^1, t^1)$ in $\mathbf{GAGraphs_{ATG^1}}$ such that (3) commutes in $\mathbf{GAGraphs_{ATG^2}}$. From pullback (1) and commutativity of (2) we obtain a unique $g^* : AG^3 \to AG^1$ in **GAGraphs** such that (4) and (5) commute.



Now $g^* : (AG^3, t^3) \to f^<(AG^2, t^2) = (AG^1, t^1)$ in $\mathbf{GAGraphs_{ATG^1}}$ means exactly commutativity of (4). Commutativity of (3) in $\mathbf{GAGraphs_{ATG^2}}$ is given by that of (5) in **GAGraphs**. This shows also the uniqueness of g^* in $\mathbf{GAGraphs_{ATG^1}}$ such that (3) commutes.

Remark 6. Note that $\mathbf{AGraphs_{ATG^1}}$ is the subcategory of $\mathbf{GAGraphs_{ATG^1}}$ consisting only of graphs (AG^1, t^1) where t^1 is signature preserving (or more precisely, t^1_S is an identity). For (AG^1, t^1) in $\mathbf{GAGraphs_{ATG^1}}$, $t^1 : AG^1 \to ATG^1$ may not be signature preserving, but for $t^1_S : DSIG^1 \to DSIG^2$ the GAG-morphism $t^1_D : D^1 \to Z_{DSIG^2}$ is given by $t^1_D : D^1 \to V_{f_S}(Z_{DSIG^2})$, where $t^1_{D,s} : D^1_s \to \{f_S(s)\}$ is uniquely determined.

In order to restrict forward typing to $f^> : \mathbf{AGraphs_{ATG^1}} \to \mathbf{AGraphs_{ATG^2}}$ we have to extend the $DSIG^1$-data type D^1 of (AG^1, t^1) to a $DSIG^2$-data type D^2 of $f^{>}(AG^1, t^1)$, where $D^2 = F_{f_S}(D^1)$ may be a suitable choice.

In the following Thms. 3 and 4 we shall consider two different cases for componentwise pushout constructions in **GAGraphs**.

Theorem 3 (Pushouts in GAGraphs over persistent morphisms). *Given persistent morphisms* $f' : AG^0 \to AG^1$ *and* $g' : AG^0 \to AG^2$ *in* **GAGraphs** *then the following construction* (1) *is a pushout in* **GAGraphs**. *Moreover, the pushout construction preserves injective, signature preserving, and persistent morphisms.*

Construction.

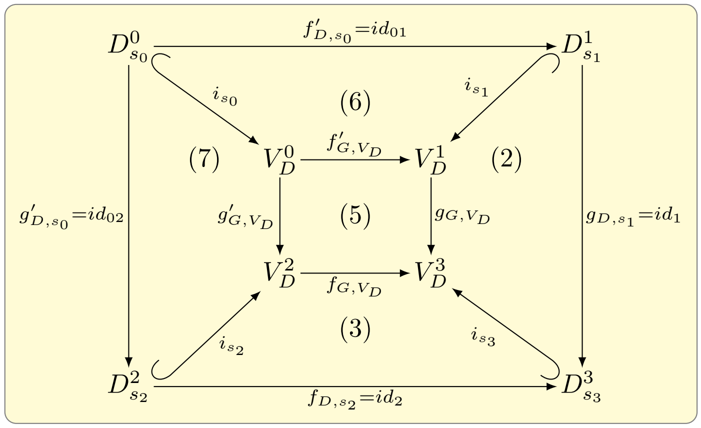
$$\begin{array}{ccc}
AG^0 = (G^0, DSIG^0, D^0) & \xrightarrow{f'=(f'_G, f'_S, f'_D)} & (G^1, DSIG^1, D^1) = AG^1 \\
{\scriptstyle g'=(g'_G, g'_S, g'_D)}\downarrow & (1) & \downarrow{\scriptstyle g=(g_G, g_S, g_D)} \\
AG^2 = (G^2, DSIG^2, D^2) & \xrightarrow[f=(f_G, f_S, f_D)]{} & (G^3, DSIG^3, D^3) = AG^3
\end{array}$$

For the G- and S-components, we have pushouts in the categories **EGraphs** and **Signatures** which are constructed componentwise in **Sets**, with attribute value sorts $S^3_D = g_s(S^1_D) \cup f_S(S^2_D)$. In the D-component we assume w.l.o.g. $f'_D = id$ and $g'_D = id$, i.e. $V_{f'_S}(D^1) = D^0 = V_{g'_S}(D^2)$, and define D_3 by amalgamation as $D_3 = D^1 +_{D^0} D^2$ with $g_D = id$ and $f_D = id$.

Proof. By construction, we have pushouts in all three components in the categories **EGraphs**, **Signatures** and **GenAlgs** of generalized algebras, and $S^3_D \subseteq S^3$. For the well-definedness of the pushout construction it remains to construct injective $i_{s_3} : D^3_{s_3} \to V^3_D$ for all $s_3 \in S^3_D$ such that the compatibility diagrams (2) and (3) commute with $s_3 = g_S(s_1)$ and $s_3 = f_S(s_2)$, respectively, and we have (4) $\dot{\bigcup}_{s_3 \in S^3_D} D^3_{s_3} = V^3_D$. Finally, the universal pushout properties have to be shown.

$$\begin{array}{ccc}
D^1_{s_1} & \xrightarrow{id} & D^3_{g_S(s_1)} \\
{\scriptstyle i_{s_1}}\downarrow & (2) & \downarrow{\scriptstyle i_{s_3}} \\
V^1_D & \xrightarrow[g_{G,V_D}]{} & V^3_D
\end{array}
\qquad
\begin{array}{ccc}
D^2_{s_2} & \xrightarrow{id} & D^3_{f_S(s_2)} \\
{\scriptstyle i_{s_2}}\downarrow & (3) & \downarrow{\scriptstyle i_{s_3}} \\
V^2_D & \xrightarrow[f'_{G,V_D}]{} & V^3_D
\end{array}$$

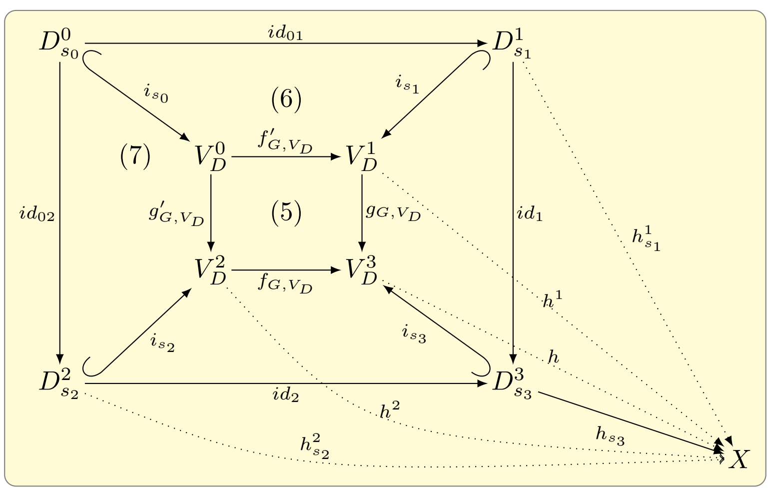
In the following diagram, let (5) be the pushout in the V_D-component of **EGraphs**, and (6) and (7) the compatibility diagrams for f' and g', respectively. We shall construct i_{s_3} such that (2) and (3) commute.



For $s_3 \in S^3_D$ we have by pushout construction in the S-component one of the following three cases:

1. There exists a unique $s_1 \in S^1_D \backslash f'_S(S^0_D)$ such that $g_S(s_1) = s_3$. Then we define $i_{s_3} = g_{G,V_D} \circ i_{s_1}$ such that (2) commutes.
2. There exists a unique $s_2 \in S^2_D \backslash g'_S(S^0_D)$ such that $f_S(s_2) = s_3$. Then we define $i_{s_3} = f_{G,V_D} \circ i_{s_2}$ such that (3) commutes.
3. There exist $s_0 \in S^0_D$ such that $(*)$ $f'_S(g_S(s_0)) = g'_S(f_S(s_0)) = s_3$. By amalgamation we have $D^0_{s_0} = D^3_{s_3}$ for all $s_0 \in S^0$ that fulfil $(*)$. In this case we define $i_{s_3} = f_{G,V_D} \circ g'_{G,V_D} \circ i_{s_0}$, and the commutativity of (6) and (7) implies that of (2) and (3), respectively.

In order to show that $i_{s_3} : D^3_{s_3} \to V^3_D$ for $s_3 \in S^3_D$ defines a coproduct in **Sets** as required in (4) we assume to have $h_{s_3} : D^3_{s_3} \to X$ for all $s_3 \in S^3_D$ and shall construct a unique $h : V^3_D \to X$ with $h \circ i_{s_3} = h_{s_3}$.



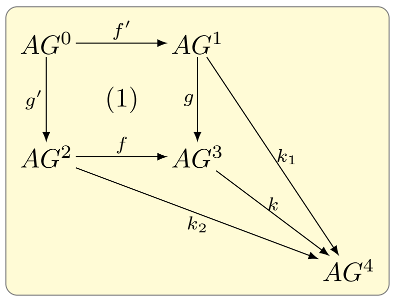
For all $s_1 \in S^1_D$ define $h^1_{s_1} = h_{s_3}$ for $s_3 = g_S(s_1)$ leading to a unique $h^1 : V^1_D \to X$ with $h^1 \circ i_{s_1} = h^1_{s_1}$. Similarly, for $s_2 \in S^2_D$ define $h^2_{s_2} = h_{s_3}$ for $s_3 = f_S(s_2)$

leading to a unique $h^2 : V_D^2 \to X$ with $h^2 \circ i_{s_2} = h^2_{s_2}$. Now $h^1 \circ f'_{G,V_D} \circ i_{s_0} = h^2 \circ g'_{G,V_D} \circ i_{s_0}$ for all $s_0 \in S_D^0$ follows from commutativity of the diagram and implies $h^1 \circ f'_{G,V_D} = h^2 \circ g'_{G,V_D}$ because V_D^0 is coproduct. This implies a unique $h : V_D^3 \to X$ with $h \circ g_{G,V_D} = h^1$ and $h \circ f_{G,V_D} = h^2$ by pushout (5).

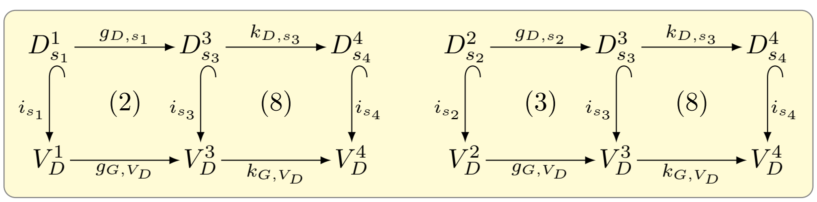
Now we have for $s_3 = g_S(s_1)$ (and similarly for $s_3 = f_S(s_2)$) that $h \circ i_{s_3} = h \circ i_{s_3} \circ id_1 = h \circ g_{G,V_D} \circ i_{s_1} = h^1 \circ i_{s_1} = h^1_{s_1} = h_{s_3}$.

In order to show the uniqueness of h let $h' \circ i_{s_3} = h_{s_3}$ for all $s_3 \in S_D^3$. For all $s_1 \in S_D^1$ we have that $h' \circ g_{G,V_D} \circ i_{s_1} = h' \circ i_{s_3} \circ id_1 = h_{s_3} = h^1_{s_1} = h^1 \circ i_{s_1}$. This implies $h' \circ g_{G,V_D} = h^1$, and similarly $h' \circ g_{G,V_D} = h^2$. Uniqueness in pushout (5) implies $h' = h$. This completes the proof of (4).

The universal property of pushout (1) in **GAGraphs** follows from that in the components, where it remains to show the commutativity of the compatibility diagram (8) for the induced morphism k.



In the case $s_3 = g_S(s_1)$ we have commutativity of (2) and of (2 + 8) by compatibility of g and k_1, respectively. This implies commutativity of (8) because $g_{D,s_1} = id_1$ is an identity. In the case $s_3 = f_S(s_2)$ a similar argument holds with (3) instead of (2).



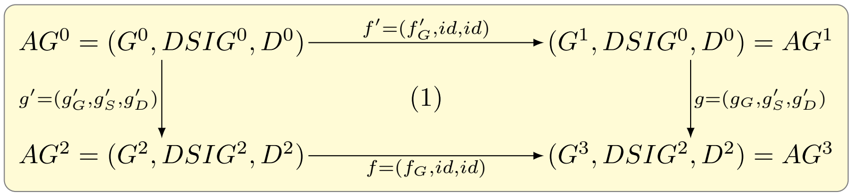
Finally, the pushout construction preserves persistent, injective, and signature preserving morphisms because in each component injective and isomorphic morphisms are preserved.

Remark 7. Given a commutative diagram (1) in **GAGraphs** as in the construction where we have pushouts in each component and f', g' persistent then the diagram (1) is a pushout in **GAGraphs**.

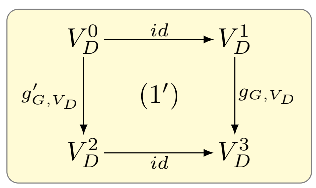
Theorem 4 (Pushouts in GAGraphs along persistent, signature preserving morphisms).

Given a persistent and signature preserving morphism $f' : AG^0 \to AG^1$ and general $g' : AG^0 \to AG^2$ in **GAGraphs** *then the following construction (1) is a pushout in* **GAGraphs**. *Moreover, the pushout construction preserves injective, signature preserving, and persistent morphisms.*

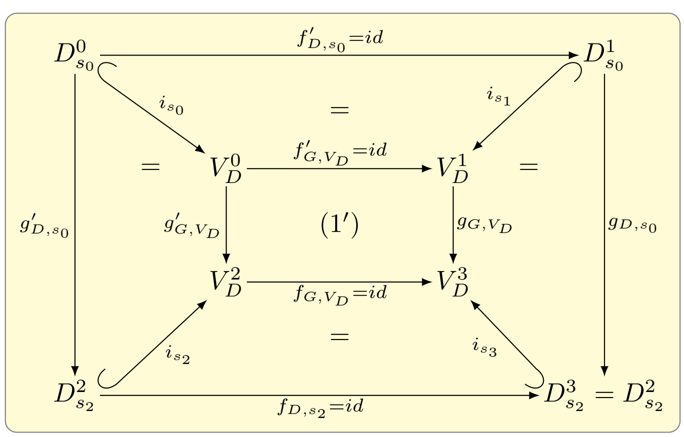
Construction. Since f' is persistent and signature perserving we assume w.l.o.g. $f'_S = id$ and $f'_D = id$ which implies that $DSIG^1 = DSIG^0$ and $D^1 = D^0$. Now let $DSIG^3 = DSIG^2$, $S^3_D = S^2_D$, $D^3 = D^2$, $f_S = id$, $f_D = id$, $g_S = g'_S$, and $g_D = g'_D$.



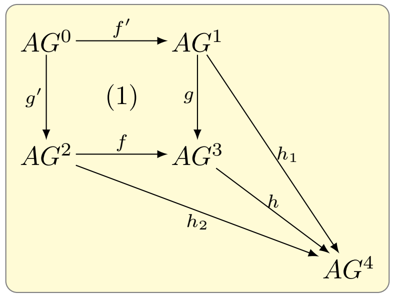
In the G-component we have a pushout in the category **EGraphs**, and in the S- and D-components we have special pushouts with identities. Compatibility of f' and g' allows to consider the following special pushout $(1')$ with identities in the V_D-component of E-Graphs.



Proof. By construction we have pushouts in all three components and $V^3_D = \dot{\bigcup}_{s_3 \in S^3_D} D^3_{s_3}$ because $V^3_D = V^2_D = \dot{\bigcup}_{s_2 \in S^2_D} D^2_{s_2} = \dot{\bigcup}_{s_3 \in S^3_D} D^3_{s_3}$ using $(1')$, $S^3_D = S^2_D$ and $D^3 = D^2$. Moreover, compatibility of f is trivial, and that of g follows from that of g' using $g_{G,V_D} = g'_{G,V_D}$, $g_S = g'_S$, and $g_D = g'_D$ as shown in the following diagram.



The universal property of pushout (1) in **GAGraphs** follows from that of the components where the compatibility for the induced morphism $h : AG^3 \to AG^4$ follows from that of h_2 using $f_{G,V_D} = id$ and $f_{D,s_2} = id$.



Remark 8. 1. Pushouts in **AGraphs** along $\mathcal{M}$-morphisms are special cases of the pushouts above, where in addition f' is injective and g' is signature preserving. Hence pushouts in **AGraphs** along $\mathcal{M}$-morphisms are also pushouts in **GAGraphs**.

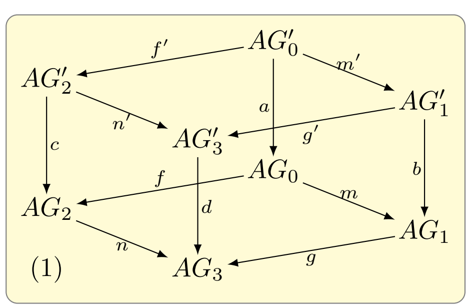
2. Given a commutative diagram (1) in **GAGraphs**, let the G-component be a pushout in **EGraphs**, and f' and f are persistent and signature preserving, then (1) is a pushout in **GAGraphs**, because then it is isomorphic to the construction in (1) where we have used identities instead of isomorphisms.
3. The pushout construction in (1) with f' persistent and signature preserving is also a pullback in the S- and D-components, because f'_S, f'_D, f_S and f_D are identities or isomorphisms.

Using the different pushout constructions in Thms. 3 and 4 we obtain two different special van Kampen properties in **GAGraphs** in Thm. 5 and can show that $(\mathbf{GAGraphs}, \mathcal{M})$ is an adhesive HLR category in Thm. 6.

Theorem 5 (Special van Kampen properties in GAGraphs).

1. *Given the following commutative cube* (1) *in* **GAGraphs**, *where* m *is persistent and injective,* f *is persistent, the bottom face is a pushout and the back faces are pullbacks in* **GAGraphs**, *then we have:*

$$\textit{the top face is a pushout} \Leftrightarrow \textit{the front faces are pullbacks.}$$



2. *Given the above commutative cube* (1) *in* **GAGraphs** *where m is persistent, injective and signature preserving, the bottom face is a pushout and the back faces are pullbacks in* **GAGraphs***, then we have:*

$$\textit{the top face is a pushout} \Leftrightarrow \textit{the front faces are pullbacks.}$$

Proof. Since $(\mathbf{EGraphs}, \mathcal{M})$ and $(\mathbf{Signatures}, \mathcal{M}_{inj})$ are adhesive HLR categories we know that the van Kampen property is valid in the G- and S-components in **GAGraphs**. It remains to show the corresponding property for the D-component.

1. By Thms. 1 and 3, persistency is preserved and we know that m, n, f, g, m' and f' are persistent.
   If the top face is a pushout then also n' and g' are persistent. Hence the front faces have opposite pairs of persistent morphisms leading to pullbacks in the D-component of these faces.
   Vice versa, given that the front faces are pullbacks we can conclude that n' and g' are persistent. Hence the amalgamation lemma for data types implies that $D'_3 = D'_1 +_{D'_0} D'_2$ which implies that we have a pushout in the D-component.
2. By Thms. 1 and 4, pushouts and pullbacks preserve persistent and signature preserving morphisms, and thus we know that m, n, and m' are persistent and signature preserving. Since either the top face is a pushout or the front left face is a pullback by precondition, also n' is persistent and signature preserving. It follows that $D_0 \stackrel{\sim}{=} D_1$, $D'_0 \stackrel{\sim}{=} D'_1$, $D_2 \stackrel{\sim}{=} D_3$ and $D'_2 \stackrel{\sim}{=} D'_3$.
   If the top face is a pushout, the front left face as a commutative diagram with an opposite pair of isomorphisms becomes a pullback. The front right face is isomorphic to the back left face and thus also a pullback.
   Vice versa, if the front faces are pullbacks, the top face is a commutative diagram with an opposite pair of isomorphisms and thus a pushout.

Theorem 6 ($(\mathbf{GAGraphs}, \mathcal{M})$ is an adhesive HLR category). *Let $\mathcal{M}$ be the class of all injective, persistent, and signature preserving morphisms in* **GAGraphs***, then* $(\mathbf{GAGraphs}, \mathcal{M})$ *is an adhesive HLR category.*

Remark 9. $\mathcal{M}$ coincides with the corresponding class in $(\mathbf{AGraphs}, \mathcal{M})$.

Proof. The van Kampen property follows directly from Part 2 of Thm. 5.

Moreover, the class $\mathcal{M}$ is closed under composition and decomposition. According to Thm. 4 we have pushouts in **GAGraphs** along $\mathcal{M}$-morphisms and $\mathcal{M}$-morphisms are closed under pushouts. According to Thm. 1 we have pullbacks in **GAGraphs** along $\mathcal{M}$-morphisms and $\mathcal{M}$-morphisms are closed under pullbacks.

Corollary 1. $(\mathbf{GAGraphs_{ATG}}, \mathcal{M})$ *is an adhesive HLR category.*

Theorem 7 (Coproduct in GAGraphs). *In* **GAGraphs** *we have general coproducts compatible with $\mathcal{M}$.*

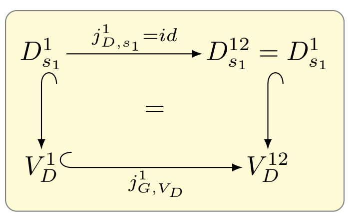
Construction. Given attributed graphs $AG^i = (G^i, DSIG^i, D^i)$ for $i = 1, 2$, then the coproduct is given by $AG^1 + AG^2 = AG^{12} = (G^1 + G^2, DSIG^1 + DSIG^2, D^{12})$, where $G^1 + G^2$ and $DSIG^1 + DSIG^2$ are the coproducts in **EGraphs** and **Signatures**, respectively, which are constructed as componentwise disjoint unions in **Sets**. For D^{12} we define $D^{12}_s = \begin{cases} D^1_s & ; s \in S^1 \\ D^2_s & ; s \in S^2 \end{cases}$ and $op_{D^{12}} = \begin{cases} op_{D^1} & ; op \in OP^1 \\ op_{D^2} & ; op \in OP^2 \end{cases}$.

The coproduct injections $j^i = (j^i_G, j^i_S, j^i_D) : AG^i \to AG^{12}$ are given by the corresponding coproduct injections $j^i_G : G^i \to G^{12}$ and $j^i_S : DSIG^i \to DSIG^{12}$, and by $j^i_D : D^i \to D^{12}$ with $j^i_{D,s} = id_{D^i_s}$, i.e. $j^i = id : D^i \to V_{j^i_S}(D^{12}) = D^i$.

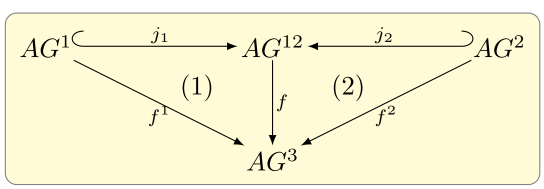
Proof. We have to show that AG^{12} is a well-defined attributed graph, that the coproduct injections are well-defined GAG-morphisms, and that the universal coproduct property is fulfilled.

1. We have to show that $\dot{\bigcup}_{s \in S^1_D + S^2_D} D^{12}_s = V^{12}_D$.

   By definition, we have $\dot{\bigcup}_{s \in S^1_D + S^2_D} D^{12}_s = \dot{\bigcup}_{s \in S^1_D} D^{12}_s \,\dot{\cup}\, \dot{\bigcup}_{s \in S^2_D} D^{12}_s = \dot{\bigcup}_{s \in S^1_D} D^1_s \,\dot{\cup}\, \dot{\bigcup}_{s \in S^2_D} D^2_s = V^1_D \,\dot{\cup}\, V^2_D = V^{12}_D$.
2. We have to show the compatibility property for the injection j^1 (and analogously for j^2). The commutativity of the following diagram is obvious since all morphisms are inclusions or identities.

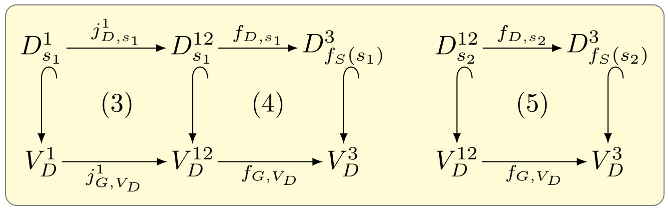


3. For the universal coproduct property, consider morphisms $f^1 : AG^1 \to AG^3$ and $f^2 : AG^2 \to AG^3$. We have to find a unique $f : AG^{12} \to AG^3$ such that (1) and (2) commute.



   Define $f = (f_G, f_S, f_D)$ with f_G induced by f^1_G and f^2_G, f_S induced by f^1_S and f^2_S, and $f_D : D^{12} \to D^3$ with $f_{D,s} = \begin{cases} f^1_{D,s} & ; s \in S^1 \\ f^2_{D,s} & ; s \in S^2 \end{cases}$ such that (1) and (2) commute.

For $s_1 \in S^1$, (3) and $(3+4)$ commute by compatibility of j^1 and f^1, respectively. Since $j^i_{D,s}$ is surjective, also (4) commutes. Analogously, for $s_2 \in S^2$, (5) commutes. This shows that f fulfils the compatibility property for all $s \in S^{12} = S^1 \dot\cup S^2$.



The uniqueness of f follows from the uniqueness of f_G and f_S and, for the D-component, from the surjectivity of $j^1_{s_1}$ for $s_1 \in S^1$ and $j^2_{s_2}$ for $s_2 \in S^2$.

4. If f^1 and f^2 are injective, also f is injective, since binary coproducts in **Sets** are compatible with injective morphisms and $f_{D,s}$ is defined by an injective f^1_s or f^2_s.

2 Type Hierarchies and Views of Visual Languages

In the metamodel approach of visual languages, a metamodel is given by an attributed type graph ATG together with structural constraints, and the corresponding visual language VL is given by all attributed graphs typed over ATG which satisfy these constraints. In the following, we study type hierarchies and views of visual languages based on morphisms in **GAGraphs**, which allows to change not only the graph structure but also the data signature and data type.

Definition 11 (Visual language). *Given an attributed type graph ATG, the* visual language VL *of ATG consists of all typed attributed graphs* $(AG, t : AG \to ATG)$ *tpyed over ATG, i.e.* $VL = Ob_{\mathbf{GAGraphs_{ATG}}}$.

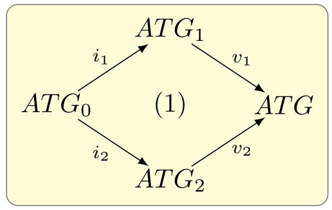
Remark 10. In contrast to $\mathbf{AGraphs_{ATG}}$, the typing $t : AG \to ATG$ in $\mathbf{GAGraphs_{ATG}}$ allows a change of the data signature from AG to ATG.

This more general concept of typing allows forward typing $f^> : \mathbf{GAGraphs_{ATG_1}} \to \mathbf{GAGraphs_{ATG_2}}$ in a straight forward way by $f^>(AG_1, t_1) = (AG_1, f \circ t_1)$ (see Def. 10). Otherwise we would have to define $f^>(AG_1, t_1) = (AG_2, f \circ t_2)$ where AG_2 is obtained by extending the $DSIG_1$-data type D_1 to a $DSIG_2$-data type D_2.

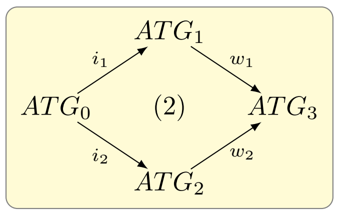
Definition 12 (Type hierarchies of visual languages). *A* type hierarchy *of visual languages VL_1 and VL_2 of attributed type graphs ATG_1 and ATG_2, respectively, is given by a GATG-morphism $f : ATG_1 \to ATG_2$.*

Definition 13 (View). *A* view *of a visual language VL over an attributed type graph ATG is given by an injective GATG-morphism $v_1 : ATG_1 \to ATG$.*

Definition 14 (Interaction and integration of views). *Given views (ATG_1, v_1) and (ATG_2, v_2) over ATG the* interaction (ATG_0, i_1, i_2) *is given by the following pullback* (1) *in* **GAGraphs***, where (ATG_0, v_0) with $v_0 = v_1 \circ i_1 = v_2 \circ i_2$ is a view over ATG and also called subview of (ATG_1, v_1) and (ATG_2, v_2).*



The integration *of views (ATG_1, v_1) and (ATG_2, v_2) with interaction (ATG_0, i_1, i_2) is given by the following pushout* (2) *in* **GAGraphs***.*

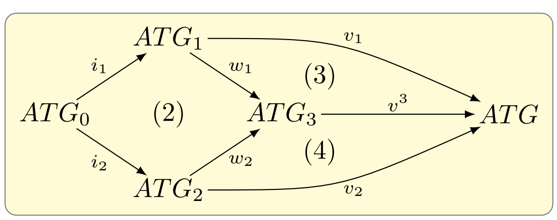


Due to the universal pushout property there is a unique injective GATG-morphism $v_3 : ATG_3 \to ATG$ such that (ATG_3, v_3) is a view over ATG.

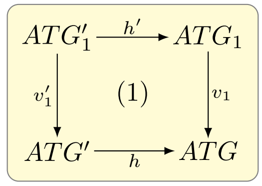
ATG is covered *by views (ATG_i, v_i) with $i \in I$ if the family $(v_i : ATG_i \to ATG)$ is jointly surjective.*

Fact 1. *If ATG is covered by views (ATG_1, v_i) for $i = 1, 2$ then the integration ATG_3 is equal to ATG up to isomorphism.*

Proof. The unique morphism v_3 with commutativity of (3) and (4) is injective in the G- and S-components due to general properties of adhesive HLR categories and injective in the D-component as a general property of GATG-morphisms. Surjectivity of v_3 follows from joint surjectivity of v_1 and v_2 in (3) and (4).



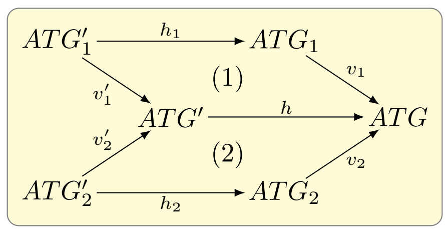
Definition 15 (Restriction of views). *Given a type hierarchy morphism $h : ATG' \to ATG$ and a view (ATG_1, v_1) over ATG then the restriction (ATG'_1, v'_1) of this view along h is defined by the following pullback* (1) *in* **GAGraphs***.*



Remark 11. Note that the restriction (ATG'_1, v'_1) is a view over ATG' because pullbacks preserve injectivity.

Fact 2. *Given a hierarchy morphism $h : ATG' \to ATG$ and views (ATG_i, v_i) for $i = 1, 2$ covering ATG, then the restrictions (ATG'_i, v'_i) along h are covering ATG'.*

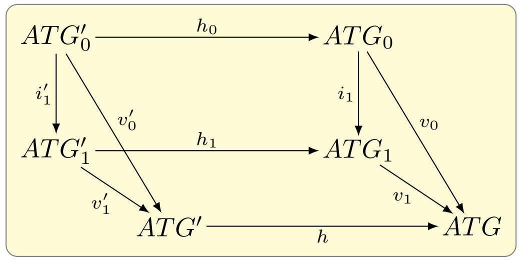
Proof. In the following diagram, v_1 and v_2 being jointly surjective implies that also v'_1 and v'_2 are jointly surjective because (1) and (2) are componentwise pullbacks.



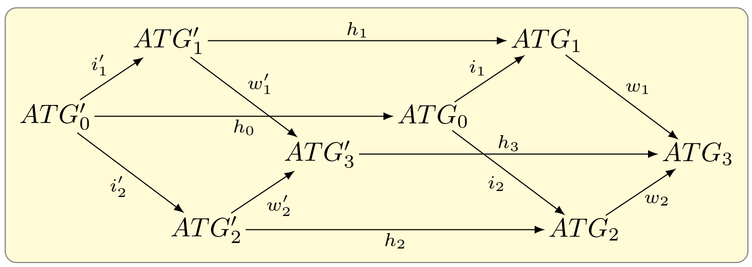
Fact 3. *Given a hierarchy morphism $h : ATG' \to ATG$ and views (ATG_i, v_i) for $i = 0, 1, 2$ over ATG with restrictions (ATG'_i, v'_i) along h and induced hierarchy morphisms $h_i : ATG'_i \to ATG_i$ then we have the following properties:*

1. *If (ATG_0, v_0) is a subview of (ATG_1, v_1) with $i_1 : ATG_0 \to ATG_1$ then also (ATG'_0, v'_0) is a subview of (ATG'_1, v'_1) with $i' : ATG'_0 \to ATG'_1$, and (ATG'_0, i'_1) is the restriction of (ATG_0, i_1) along h_1.*
2. *If (ATG_3, v_3) is the integration of (ATG_i, v_i) for $i = 1, 2$ with interaction (ATG_0, i_1, i_2) then also the restriction (ATG'_3, v'_3) is the integration of the restrictions (ATG_i, v_i) for $i = 1, 2$ with an interaction (ATG'_0, i'_1, i'_2).*

Proof. 1. In the following diagram, the right triangle commutes because (ATG_0, v_0) is a subview of (ATG_1, v_1) with injective $i_1 : ATG_0 \to ATG_1$. The bottom and diagonal squares are pullbacks by the definition of the restrictions (ATG'_1, v'_1) and (ATG'_0, v'_0). The pullback property implies that there exists a unique injective $i'_1 : ATG'_0 \to ATG'_1$ such that the left triangle and the back square commute. Moreover, the back square is a pullback by pullback decomposition which shows that (ATG'_0, i'_1) is the restriction of (ATG_0, i_1) along $h_1 : ATG'_1 \to ATG_1$.

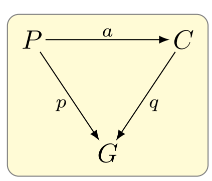


2. In the following cube, the right face is a pushout because (ATG_3, v_3) is the integration of (ATG_i, v_i) for $i = 1, 2$ with the interaction (ATG_0, i_1, i_2). The front and back faces are pullbacks due to Part 1. Since all morphisms in the right and left faces are injective and also persistent as GATG-morphisms we can apply Part 1 of Thm. 5 to conclude that also the left face is a pushout and hence (ATG'_3, v'_3) is the integration of (ATG'_i, v'_i) with $i = 1, 2$ with the interaction (ATG'_0, v'_0).



2.1 Type Hierarchies and Views with Constraints

In this subsection we consider visual languages VL given by an attributed type graph ATG with a set of constraints PC, where the visual language is defined by $VL = \{G \in \mathbf{GAGraphs_{ATG}} \mid G \models c \ \forall c \in PC\}$. A constraint $c = ((P, t_P) \xrightarrow{a} (C, t_C))$ is given by typed attributed graphs (P, t_P) and (C, t_C) typed over ATG, where we omit the typing morphisms if they are not necessary, i.e. write $c = (P \xrightarrow{a} C)$, and a typed attributed graph morphism $a : P \to C$. A typed attributed graph G typed over ATG fulfills a constraint $c = (P \xrightarrow{a} C) \in PC$ iff for all typed attributed graph morphisms $p : P \to G$ there exists an injective $q : C \to G$ such that $q \circ a = p$.



Definition 16 (Forward translation of constraints). *Given a GATG-morphism $f : ATG_1 \to ATG_2$ and a constraint $c_1 = ((P, t_P) \xrightarrow{a} (C, t_C))$*

over ATG_1*, the* forward translated constraint $f^>(c_1) = c_2$ *over* ATG_2 *is given by* $c_2 = ((P, f \circ t_P) \xrightarrow{a} (C, f \circ t_C))$.

For a set PC_1 *of constraints over* ATG_1 *define* $f^>(PC_1) = \{f^>(c_1) \mid c_1 \in PC_1\}$.

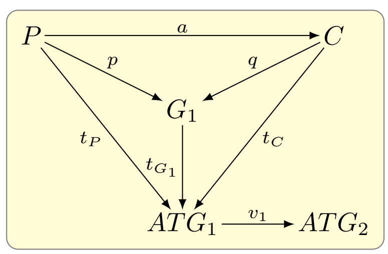
Fact 4. *Given a view* (ATG_1, v_1) *over* ATG_2*, a constraint* $c_1 \in PC_1$ *typed over* ATG_1*, and a typed attributed graph* G_1 *typed over* ATG_1*, then we have:*

$$G_1 \models c_1 \Leftrightarrow v_1^>(G_1) \models v_1^>(c_1),$$

where $v_1^>(G_1)$ *and* $v_1^>(c_1)$ *are the corresponding forward translations over* ATG_2*.*

Proof. For $c_1 = ((P, t_P) \xrightarrow{a} (C, t_C))$ we have $v_1^>(c_1) = ((P, v_1 \circ t_P) \xrightarrow{a} (C, v_1 \circ t_C))$, and $v_1^>(G_1, t_{G_1}) = (G_1, v_1 \circ t_{G_1})$.

$\Rightarrow$ We have to show that for each injective $p : P \to G_1$ in $\mathbf{GAGraphs_{ATG_2}}$ there is an injective $q : C \to G_1$ in $\mathbf{GAGraphs_{ATG_2}}$ with $q \circ a = p$. Given an injective $p : P \to G_1$ in $\mathbf{GAGraphs_{ATG_2}}$ we have $p : P \to G_1$ in **GAGraphs** with $v_1 \circ t_P = v_1 \circ t_{G_1} \circ p$. Since v_1 is injective it follows that $t_P = t_{G_1} \circ p$, i.e. p is also an $\mathbf{GAGraphs_{ATG_1}}$-morphism. Since $G_1 \models c_1$ there exists an injective $q : C \to G_1$ with $q \circ a = p$ in $\mathbf{GAGraphs_{ATG_1}}$, i.e. $t_{G_1} \circ q = t_C$. Hence $v_1 \circ t_{G_1} \circ q = v_1 \circ t_C$ and q is the required $\mathbf{GAGraphs_{ATG_2}}$-morphism.



$\Leftarrow$ We have to show that for each injective $p : P \to G_1$ in $\mathbf{GAGraphs_{ATG_1}}$ there is an injective $q : C \to G_1$ in $\mathbf{GAGraphs_{ATG_1}}$ with $q \circ a = p$. Given an injective $p : P \to G_1$ in $\mathbf{GAGraphs_{ATG_1}}$ we have $p : P \to G_1$ in **GAGraphs** with $t_P = t_{G_1} \circ p$. With $v_1 \circ t_P = v_1 \circ t_{G_1} \circ p$, p is also a $\mathbf{GAGraphs_{ATG_2}}$-morphism. Since $v_1^>(G_1) \models v_1^>(c_1)$ there exists an injective $q : C \to G_1$ with $q \circ a = p$ in $\mathbf{GAGraphs_{ATG_2}}$, i.e. $v_1 \circ t_{G_1} \circ q = v_1 \circ t_C$. Since v_1 is injective it follows that $t_{G_1} \circ q = t_C$, hence q is the required $\mathbf{GAGraphs_{ATG_1}}$-morphism.

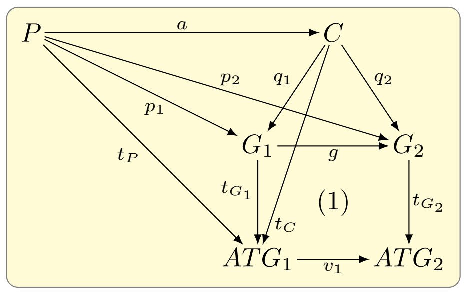
Fact 5. *Given a view* (ATG_1, v_1) *over* ATG_2*, a constraint* $c_1 \in PC_1$ *typed over* ATG_1*, and a typed attributed graph* G_2 *typed over* ATG_2*, then we have:*

$$G_2 \models v_1^>(c_1) \Leftrightarrow v_1^<(G_2) \models c_1,$$

where $v_1^>(c_1)$ *is the forward translation of* c_1 *and* $v_1^<(G_2)$ *is the backward translation of* G_2*.*

Proof. For $c_1 = ((P, t_P) \xrightarrow{a} (C, t_C))$ we have $v_1^>(c_1) = ((P, v_1 \circ t_P) \xrightarrow{a} (C, v_1 \circ t_C))$, and $v_1^<(G_2, t_{G_2}) = (G_1, t_{G_1})$ with pullback (1).

$\Rightarrow$ We have to show that for each injective $p_1 : P \to G_1$ in $\mathbf{GAGraphs_{ATG_1}}$ there is an injective $q_1 : C \to G_1$ in $\mathbf{GAGraphs_{ATG_1}}$ with $q_1 \circ a = p_1$. Given an injective $p_1 : P \to G_1$ in $\mathbf{GAGraphs_{ATG_1}}$, with v_1 being injective and (1) being a pullback also g and hence $g \circ p_1$ are injective. Thus we have that $t_{G_2} \circ g \circ p_1 = v_1 \circ t_{G_1} \circ p_1 = v_1 \circ t_P$ and since $G_2 \models v_1^>(c_1)$ there exists an injective $q_2 : C \to G_2$ with $q_2 \circ a = g \circ p_1$ in $\mathbf{GAGraphs_{ATG_2}}$, i.e. $t_{G_2} \circ q_2 = v_1 \circ t_C$. Now pullback (1) implies a unique $q_1 : C \to G_1$ with $t_{G_1} \circ q_1 = t_C$ and $g \circ q_1 = q_2$. The latter implies that q_1 is injective by decomposition of monomorphisms. Hence q_1 is the required $\mathbf{GAGraphs_{ATG_1}}$-morphism.



$\Leftarrow$ We have to show that for each injective $p_2 : P \to G_2$ in $\mathbf{GAGraphs_{ATG_2}}$ there is an injective $q_2 : C \to G_2$ in $\mathbf{GAGraphs_{ATG_2}}$ with $q_2 \circ a = p_2$. Given an injective $p_2 : P \to G_2$ in $\mathbf{GAGraphs_{ATG_2}}$ we have $t_{G_2} \circ p_2 = v_1 \circ t_P$. Pullback (1) implies a unique $p_1 : P \to G_1$ with $t_{G_1} \circ p_1 = t_P$ and $g \circ p_1 = p_2$. The latter implies that p_1 is injective by decomposition of monomorphisms. Since $G_1 \models c_1$ there exists an injective $q_1 : C \to G_1$ with $q_1 \circ a = p_1$ in $\mathbf{GAGraphs_{ATG_1}}$, i.e. $t_{G_1} \circ q_1 = t_C$. It follows that $g \circ q_1$ is injective. Thus we have that $t_{G_2} \circ g \circ q_1 = v_1 \circ t_{G_1} \circ q_1 = v_1 \circ t_C$. Hence $q_2 = g \circ q_1$ is the required $\mathbf{GAGraphs_{ATG_2}}$-morphism with $q_2 \circ a = g \circ q_1 \circ a = g \circ p_1 = p_2$.

Fact 6. *Given attributed type graphs ATG_1 and ATG_2, constraints PC_1 and PC_2 over ATG_1 and ATG_2 leading to visual languages VL_1 and VL_2, respectively, and a view (ATG_1, v_1) over ATG_2, then we have the following results:*

1. *If $v_1^>(PC_1) \Rightarrow PC_2$ then $v_1^>(G_1) \in VL_2$ for all $G_1 \in VL_1$, i.e. $v_1^> : VL_1 \to VL_2$.*
2. *If $PC_2 \Rightarrow v_1^>(PC_1)$ then $v_1^<(G_2) \in VL_1$ for all $G_2 \in VL_2$, i,e, $v_1^< : VL_2 \to VL_1$.*

Proof. 1. Given $G_1 \in VL_1$ this means that $G_1 \models PC_1$. Now Fact 4 implies that $v_1^>(G_1) \models v_1^>(PC_1)$ and if $v_1^>(PC_1) \Rightarrow PC_2$ also $v_1^>(G_1) \models PC_2$, i.e. $v_1^>(G_1) \in VL_2$.
2. Given $G_2 \in VL_2$ this means that $G_2 \models PC_2$, and if $PC_2 \Rightarrow v_1^>(PC_1)$ also $G_2 \models v_1^>(PC_1)$. Now Fact 5 implies that $v_1^<(G_2) \models PC_1$, i.e. $v_1^<(G_2) \in VL_1$.

Definition 17 (View with constraints). *Given attributed type graphs ATG_1 and ATG_2, constraints PC_1 and PC_2 over ATG_1 and ATG_2, respectively, and a view (ATG_1, v_1) over ATG_2, then (ATG_1, v_1) is a* view with constraints *if $PC_2 \Rightarrow v_1^{>}(PC_1)$.*

(ATG, PC) is covered *by views with constraints (ATG_1, PC_1, v_1) and (ATG_2, PC_2, v_2) if ATG is covered by (ATG_1, v_1) and (ATG_2, v_2), and $PC = v_1^{>}(PC_1) \cup v_2^{>}(PC_2)$.*

3 Models and View-Models of Visual Languages

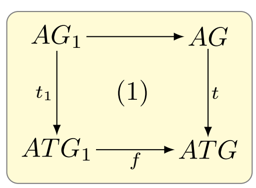
In this section we study models of visual languages and of views of visual languages, called view-models.

Definition 18 (Model). *Given a metamodel of a visual language VL by an attributed type graph ATG then a* model *of VL is a typed attributed graph AG typed over ATG, where the typing $t : AG \to ATG$ is a GAG-morphism.*

The model (AG, t) is called signature-conform if t is signature-preserving.

All models of ATG define the category $\mathbf{GAGraphs_{ATG}}$, while all signature conform models define the category $\mathbf{AGraphs_{ATG}}$ which is the subcategory of $\mathbf{GAGraphs_{ATG}}$ where all morphisms are signature preserving. In the following we consider both kinds of models.

Definition 19 (Restriction). *Given a view $f : ATG_1 \to ATG$, i.e. an injective GATG-morphism, and an ATG-model (AG, t) then the* restriction (AG_1, t_1) *of (AG, t) to the view (ATG_1, f) is defined by the following pullback* (1)*, written $f^{<}(AG, t) = (AG_1, t_1)$.*



The construction $f^{<} : \mathbf{GAGraphs_{ATG}} \to \mathbf{GAGraphs_{ATG_1}}$ is called backward typing (see Def. 10). Backward typing can be restricted to signature conform models $f^{<} : \mathbf{AGraphs_{ATG}} \to \mathbf{AGraphs_{ATG_1}}$, because signature preservation of t implies that of t_1.

Definition 20 (Extension). *Given a view $f : ATG_1 \to ATG_2$ the* extension *of a view-model (AG_1, t_1) along f is given by $(AG_1, f \circ t_1)$, written $f^{>}(AG_1, t_1) = (AG_1, f \circ t_1)$.*

The construction $f^{>} : \mathbf{GAGraphs_{ATG_1}} \to \mathbf{GAGraphs_{ATG}}$ is called forward typing (see Def. 10). According to Thm. 2, we have adjoint functors

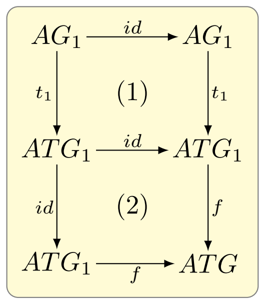
$$f^{>} \dashv f^{<} : \mathbf{GAGraphs_{ATG^2}} \to \mathbf{GAGraphs_{ATG^1}}$$

which implies that $f^>$ preserves pushouts and $f^<$ preserves pullbacks. Note that forward typing cannot be restricted to signature conform models unless f is signature conform.

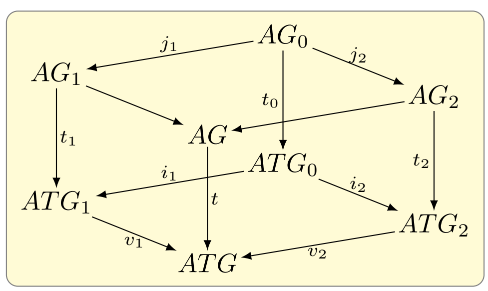
Fact 7. *Backward typing $f^<$ is left inverse to forward typing $f^>$, i.e. $f^< \circ f^> = id$.*

Altogether, forward typing is a persistent free construction w.r.t. backward typing.

Proof. For (AG_1, t_1) we have $f^>(AG_1, t_1) = (AG_1, f \circ t_1)$ and $f^<(AG_1, f \circ t_1) = (AG_1, t_1)$ according to the combined pullback $(1+2)$. Actually, (1) is a trivial pullback and (2) is a pullback because f is injective.



Definition 21 (Consistency and integration). *Given views (ATG_i, v_i) for $i = 1, 2$ of ATG with interaction (ATG_0, i_1, i_2) defined by the pullback in the bottom face of the following cube, then models (AG_i, t_i) with $i = 1, 2$ of the views (ATG_i, v_i) are called* consistent *if there is a model (AG_0, t_0) of ATG_0 such that the back faces are pullbacks, i.e. $i_1^<(AG_1, t_1) = (AG_0, t_0) = i_2^<(AG_2, t_2)$.*



A model (AG, t) of ATG is called integration *(or* amalgamation*) of consistent (AG_1, t_1) and (AG_2, t_2) via (AG_0, t_0) if the front faces of the above cube are pullbacks, i.e. $v_1^<(AG, t) = (AG_1, t_1)$ and $v_2^<(AG, t) = (AG_2, t_2)$, and the top face commutes.*

Theorem 8 (Integration and decomposition of models).
Integration. *Let ATG be covered by the views (ATG_i, v_i) for $i = 1, 2$. If (AG_i, t_i) are consistent models of (ATG_i, v_i) via (AG_0, t_0) then there is up to isomorphism a unique integration (AG, t) of (AG_i, t_i) via (AG_0, t_0).*

Decomposition. *Vice versa, each model* (AG, t) *of* ATG *can be decomposed uniquely into view-models* (AG_i, t_i) *with* $i = 1, 2$ *such that* (AG, t) *is the integration of* (AG_1, t_1) *and* (AG_2, t_2) *via* (AG_0, t_0)*.*

Bijective Correspondence. *Integration and decomposition are inverse to each other up to isomorphism.*

Proof. Integration. Since ATG is covered by (ATG_i, v_i) for $i = 1, 2$ it is also the integration of these views by Fact 1. This means that the bottom pullback is already a pushout in **GAGraphs** with injective and persistent morphisms. Now assume that (AG_i, t_i) with $i = 1, 2$ are consistent models. This means that the back faces of the cube are pullbacks with injective and persistent j_1 and j_2. This allows to construct AG in the top face as pushout in **GAGraphs** leading to a unique t such that the front faces commute. According to the van Kampen property in Part 1 of Thm. 5 the front faces are pullbacks such that (AG, t) is the integration of (AG_i, t_i) for $i = 1, 2$ via (AG_0, t_0). In order to show the uniqueness let also $(AG', t' : AG' \to ATG)$ be an integration of (AG_i, t_i) for $i = 1, 2$ via (AG_0, t_0). Then the front faces are pullbacks with (AG', t') and the top face commutes. Now the van Kampen property in the opposite direction implies that the top face is a pushout in **GAGraphs**. This implies that (AG, t) and (AG', t') are equal up to isomorphism.

Decomposition. Vice versa, given a model (AG, t) of ATG we construct the front and one of the back faces as pullbacks such that the remaining back face also becomes a pullback and the top face commutes. This shows that (AG_1, t_1) and (AG_2, t_2) are consistent w.r.t (AG_0, t_0), and (AG, t) is the integration of both via (AG_0, t_0). The decomposition is unique up to isomorphism because the pullbacks in the front faces are unique up to isomorphism.

Bijective Correspondence. Uniqueness of integration and decomposition as shown above implies that both constructions are inverse to each other up to isomorphism.

Theorem 9 (Integration and decomposition with constraints). *Let* (ATG, PC) *be covered by the views* (ATG_i, PC_i, v_i) *for* $i = 1, 2$*. If* $(AG_i, t_i) \models PC_i$ *are consistent models of* (ATG_i, v_i) *via* (AG_0, t_0) *then we have for the integration* (AG, t) *that* $AG \models PC$*.*

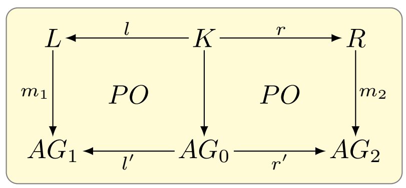
Vice versa, for the decomposition of (AG, t) *into view-models* (AG_i, t_i) *with* $i = 1, 2$ *it holds that* $AG_i \models PC_i$*.*

Proof. Given the integration (AG, t) we have that $v_1^<(AG, t) = (AG_1, t_1) \models PC_1$. Now Fact 5 shows that this is equivalent to the fact that $AG \models v_1^>(PC_1)$. Analogously we have that $AG \models v_2^>(PC_1)$, and altogether $AG \models PC$ because $PC = v_1^>(PC_1) \cup v_2^>(PC_2)$ by Def. 17.

Vice versa, $AG \models PC$ implies $AG \models v_i^>(PC_i)$ for $i = 1, 2$ and hence $AG_i = v_i^<(AG) \models PC_i$ by Fact 5.

4 Generalized Typed Graph Transformation Systems

According to Thm. 6, the category $(\mathbf{GAGraphs}, \mathcal{M})$ with the class $\mathcal{M}$ of all injective, persistent, and signature preserving morphisms and also the corresponding typed variant $(\mathbf{GAGraphs_{ATG}}, \mathcal{M})$ are adhesive HLR categories. This allows us to apply main parts of the theory for typed attributed graph transformations developed on the basis of the categories $(\mathbf{AGraphs}, \mathcal{M})$ and $(\mathbf{AGraphs_{ATG}}, \mathcal{M})$, respectively, also to the generalized case. The main difference is that graphs in $\mathbf{GAGraphs_{ATG}}$ allow for the typing $t : AG \to ATG$ a change of the data type signature. The productions $p = (L \xleftarrow{l} K \xrightarrow{r} R)$ with $l, r \in \mathcal{M}$ are the same, but the double pushout transformations in $(\mathbf{GAGraphs}, \mathcal{M})$ and $(\mathbf{GAGraphs_{ATG}}, \mathcal{M})$ allow matches m_1 and comatches m_2 with change of the signature. But $l, r \in \mathcal{M}$ implies $l', r' \in \mathcal{M}$ such that AG_1, AG_0 and AG_2 have, up to isomorphism, the same data signature and data type.



Definition 22 (Generalized typed graph transformation system). *A generalized typed graph transformation system $GTGTS = (ATG, P, \pi)$ consists of an attributed type graph ATG, a set of production names P and a mapping π that assigns to each $p \in P$ a production $\pi(p) = (L_p \xleftarrow{l_p} K_p \xrightarrow{r_p} R_p)$ with $l_p, r_p \in \mathcal{M}$.*

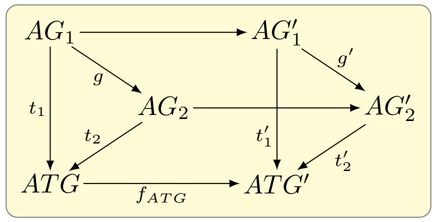
Definition 23 (GTGTS-embedding). *Given generalized typed graph transformation systems $GTGTS = (ATG, P, \pi)$ and $GTGTS' = (ATG', P', \pi')$, a* GTGTS-embedding *$f = (f_{ATG}, f_P) : GTGTS \to GTGTS'$ consists of an injective GATG-morphism $f_{ATG} : ATG \to ATG'$ and a mapping $f_P : P \to P'$ such that for each $p \in P$ we have $\pi(p) = f^{<}_{ATG}(\pi'(f_P(p)))$.*

Note that we do not require that $\pi(P)$ consists of all restrictions of productions $\pi'(P')$. In this case, f is called full GTGTS-embedding.

Remark 12. Given a forward embedding f, i.e. $f^{>}(\pi(P)) \subseteq \pi'(P')$, then f is also a GTGTS-embedding, because $\pi(P) = f^{<} \circ f^{>}(\pi(P)) \subseteq f^{<}(\pi'(P'))$. By forward typing each direct transformation $G \overset{p}{\Longrightarrow} H$ in $GTGTS$ becomes a transformation $f^{>}(G) \overset{f^{>}(p)}{\Longrightarrow} f^{>}(H)$ in $GTGTS'$ because forward typing as a left adjoint functor preserves pushouts.

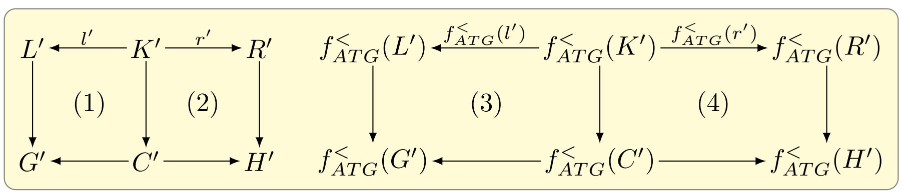
Fact 8. *Backward typing $f^{<}_{ATG}$ and hence GTGTS-embeddings preserve injective, persistent, and signature preserving morphisms, respectively, and hence also the class $\mathcal{M}$.*

Proof. Given $g' : (AG'_1, t'_1) \to (AG'_2, t'_2)$ in $\mathbf{GAGraphs_{ATG}}$ then $f^{<}_{ATG}(g') = g$ defined by the following pullbacks in **GAGraphs**.

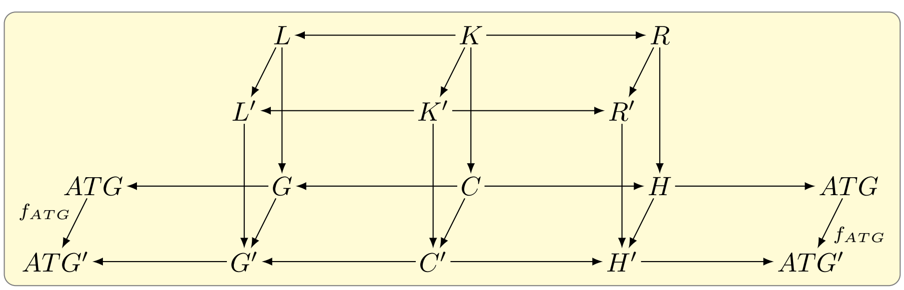


Pullbacks in **GAGraphs** preserve injective, persistent, and signature preserving morphisms, respectively.

Theorem 10 (Reflection of behaviour by GTGTS-embeddings). *Given a GTGTS-embedding* $f = (f_{ATG}, f_P) : GTGTS \to GTGTS'$ *and a direct transformation* $G' \overset{\pi'(f_P(p))}{\Longrightarrow} H'$ *in* $GTGTS'$ *by pushouts* (1) *and* (2) *via* $\pi'(f_P(p)) = (L' \overset{l'}{\longleftarrow} K' \overset{r'}{\longrightarrow} R')$ *then backwards typing leads to a direct transformation in* $GTGTS$ *by pushouts* (3) *and* (4) *via* $\pi(p) = f^{<}_{ATG}(\pi'(f_P(p)))$.



Proof. We have to show that (3) and (4) are pushouts in $\mathbf{GAGraphs_{ATG}}$. Given pushouts (1) and (2), and $f_{ATG} : ATG \to ATG'$ injective we construct G, L, K, C, H and R as restrictions of G', L', K', C', H' and R', respectively, leading to the following diagram where we have pushouts in the front faces and all the other faces except the back faces are pullbacks in **GAGraphs**.



In the left cube, the front face is the pushout (1) with $l' : K' \to L' \in \mathcal{M}$. This allows us to apply the van Kampen property in Part 2 of Thm. 5 to conclude that also the back face is a pushout in **GAGraphs**.

Similarly, in the right cube the front face is the pushout (2) with $r' : K' \to R' \in \mathcal{M}$ which implies that the back face is a pushout in **GAGraphs**.

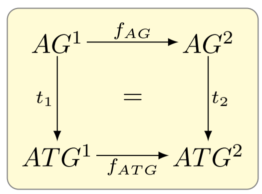
These back faces correspond exactly to the diagrams (3) and (4). Hence (3) and (4) are pushouts defining a direct transformation in $GTGTS$.

If f is a full GTGTS-embedding, by backward typing each direct transformation in $GTGTS'$ leads to a direct transformation in $GTGTS$.

References

[1] Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs. Springer (2006)

Definition 24 (Generalized typed attributed graph morphism). *Given typed attributed graphs* $TAG^i = (AG^i, t^i : AG^i \to ATG^i)$ *for* $i = 1, 2$*, a generalized typed attributed graph morphisms* $f = (f_{AG}, f_{ATG}) : TAG^1 \to TAG^2$ *is given by a GAG-morphism* $f_{AG} : AG^1 \to AG^2$ *and a GATG-morphism* $f_{ATG} : ATG^1 \to ATG^2$ *such that* $f_{ATG} \circ t^1 = t^2 \circ f_{AG}$*.*



Definition 25 (Category GTAGraphs). *Typed attributed graphs and generalized typed attributed graph morphisms form the category* **GTAGraphs***.*

Remark 13. **GTAGraphs** $\cong$ $ComCat(F, G; \{1\})$ with $F = ID$: **GAGraphs** $\to$ **GAGraphs**, $G = Inc$: **GAGraphs**$|_T$ $\to$ **GAGraphs**, where **GAGraphs**$|_T$ is the subcategory of **GAGraphs** containing all attributed type graphs.