Satisfaction, Restriction and Amalgamation of Constraints in the Framework of $\mathcal{M}$-Adhesive Categories

Hanna Schölzel[1], Hartmut Ehrig[1], Maria Maximova[1], Karsten Gabriel[1], and Frank Hermann[1,2]

1) Institut für Softwaretechnik und Theoretische Informatik, Technische Universität Berlin, Germany
2) University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust

[hannas, ehrig, mascham, kgabriel, frank]@cs.tu-berlin.de

Application conditions for rules and constraints for graphs are well-known in the theory of graph transformation and have been extended already to $\mathcal{M}$-adhesive transformation systems. According to the literature we distinguish between two kinds of satisfaction for constraints, called general and initial satisfaction of constraints, where initial satisfaction is defined for constraints over an initial object of the base category. Unfortunately, the standard definition of general satisfaction is not compatible with negation in contrast to initial satisfaction.

Based on the well-known restriction of objects along type morphisms, we study in this paper restriction and amalgamation of application conditions and constraints together with their solutions. In our main result, we show compatibility of initial satisfaction for positive constraints with restriction and amalgamation, while general satisfaction fails in general.

Our main result is based on the compatibility of composition via pushouts with restriction, which is ensured by the horizontal van Kampen property in addition to the vertical one that is generally satisfied in $\mathcal{M}$-adhesive categories.

1 Introduction

The framework of $\mathcal{M}$-adhesive categories has been introduced recently [8, 3] as a generalization of different kinds of high level replacement systems based on the double pushout (DPO) approach [6]. Prominent examples that fit into the framework of $\mathcal{M}$-adhesive categories are (typed attributed) graphs [6, 19] and (high-level) Petri nets [2, 10]. In the context of domain specific languages and model transformations based on graph transformation, graph conditions (constraints) are already used extensively for the specification of model constraints and the specification of application conditions of transformation rules. Graph conditions can be nested, may contain Boolean expressions [13, 14] and are expressively equivalent to first-order formulas on graphs [4] as shown in [14, 20]. We generally use the term "nested condition" whenever we refer to the most general case.

Restriction is a general concept for the definition of views of domain languages and is used for reducing the complexity of a model and for increasing the focus to relevant model element types. A major research challenge in this field is to provide general results that allow for reasoning on properties of the full model (system) by analyzing restricted properties on the views (restrictions) of the model only. Technically, a restriction of a model is given as a pullback along type morphisms. While this construction can be extended directly to restrictions of nested conditions, the satisfaction of the restricted nested conditions is not generally guaranteed for the restricted models, but—as we show in this paper—can be ensured under some sufficient conditions.

According to the literature [14, 6], we distinguish between two kinds of satisfaction for nested conditions, called general and initial satisfaction, where initial satisfaction is defined for nested conditions over

U. Golas, T. Soboll (Eds.): Proceedings of ACCAT 2012
EPTCS 93, 2012, pp. 83–104, doi:10.4204/EPTCS.93.5

an initial object of the base category. Intuitively, general satisfaction requires that a property holds for all occurrences of a premise pattern, while initial satisfaction requires this property for at least one occurrence. Unfortunately, the standard definition of general satisfaction is not compatible with the Boolean operators for negation and disjunction, but initial satisfaction is compatible with all Boolean operators (see App. A in [21]). In order to show, in addition, compatibility of initial satisfaction with restriction, we introduce the concept of amalgamation for typed objects, where objects can be amalgamated along their overlapping according to the given type restrictions.

As the main technical result, we show that solutions for nested conditions can be composed and decomposed along an amalgamation of them (Thm. 4.10), if the nested conditions are positive, i.e., they contain neither a negation nor a "for all" expression (universal quantification). Based on this property, we show in our main result (Thm. 5.1), that initial satisfaction of positive nested conditions is compatible with amalgamation based on restrictions that agree on their overlappings. Note in particular that this result does not hold for general satisfaction which we illustrate by a concrete counterexample.

The structure of the paper is as follows. Section 2 reviews the general framework of $\mathcal{M}$-adhesive categories and main concepts for nested conditions and their satisfaction. Thereafter, Sec. 3 presents the restriction of objects and nested conditions along type object morphisms. Section 4 contains the constructions and results concerning the amalgamation of objects and nested conditions and in Sec. 5, we present our main result showing the compatibility of initial satisfaction with amalgamation and restriction. Related work is discussed in Sec. 6. Section 7 concludes the paper and discusses aspects of future work. Appendix A contains the proofs that are not contained in the main part. Additionally, App. A in [21] provides formal details concerning the transformation between both satisfaction relations and, moreover, their compatibility resp. incompatibility with Boolean operators.

2 General Framework and Concepts

In this section we recall some basic well-known concepts and notions and introduce some new notions that we are using in our approach. Our considerations are based on the framework of $\mathcal{M}$-adhesive categories. An $\mathcal{M}$-adhesive category [8] consists of a category **C** together with a class $\mathcal{M}$ of monomorphisms as defined in Def. 2.1 below. The concept of $\mathcal{M}$-adhesive categories generalizes that of adhesive [17], adhesive HLR [9], and weak adhesive HLR categories [6].

Definition 2.1 ($\mathcal{M}$-Adhesive Category)**.** *An $\mathcal{M}$-adhesive category $(\mathbf{C},\mathcal{M})$ is a category* **C** *together with a class $\mathcal{M}$ of monomorphisms satisfying:*

- *the class $\mathcal{M}$ is closed under isomorphisms, composition and decomposition,*
- **C** *has pushouts and pullbacks along $\mathcal{M}$-morphisms,*
- *$\mathcal{M}$-morphisms are closed under pushouts and pullbacks, and*
- *it holds the vertical van Kampen (short VK) property. This means that pushouts along $\mathcal{M}$-morphisms are $\mathcal{M}$-VK squares, i. e., pushout (1) with $m \in \mathcal{M}$ is an $\mathcal{M}$-VK square, if for all commutative cubes (2) with (1) in the bottom, all vertical morphisms $a,b,c,d \in \mathcal{M}$ and pullbacks in the back faces we have that the top face is a pushout if and only if the front faces are pullbacks.*

$$
\begin{tikzcd}
& A \arrow[dl] \arrow[dr, "m"] \arrow[dd, phantom, "(1)"] & \\
C \arrow[dr] & & B \arrow[dl] \\
& D &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
& & A' \arrow[dll] \arrow[dr] \arrow[ddd, dotted, "a"] & \\
C' \arrow[dr] \arrow[ddd, "c"'] & & & B' \arrow[dll] \arrow[ddd, "b"] \\
& D' \arrow[ddd, "d"] & & \\
& & A \arrow[dll, dotted] \arrow[dr, dotted, "m"] \arrow[dl, phantom, "(1)"] & \\
C \arrow[dr] & & & B \arrow[dll] \\
& D & &
\end{tikzcd}
\qquad (2)
$$

Remark 2.2. *In Sec. 3, Sec. 4 and Sec. 5 we will also need the horizontal VK property, where the VK property is only required for commutative cubes with all horizontal morphisms in $\mathcal{M}$ (see [8]), to show the compatibility of object composition and the corresponding restrictions. Note moreover, that an $\mathcal{M}$-adhesive category which also satisfies the horizontal VK property is a weak adhesive HLR category [6].*

A set of transformation rules over an $\mathcal{M}$-adhesive category according to the DPO approach constitutes an $\mathcal{M}$-adhesive transformation system [8]. For various examples (graphs, Petri nets, etc.) see [6].

In Sec. 3, Sec. 4 and Sec. 5 we are considering $\mathcal{M}$-adhesive categories with effective pushouts. According to [18], the formal definition is as follows.

Definition 2.3 (Effective Pushout)**.** *Given $\mathcal{M}$-morphisms $a : B \to X$, $b : C \to X$ in an $\mathcal{M}$-adhesive category $(\mathbf{C}, \mathcal{M})$ and let (A, p_1, p_2) be obtained by the pullback of a and b. Then pushout (1) of p_1 and p_2 is called effective, if the unique morphism $u : D \to X$ induced by pushout (1) is an $\mathcal{M}$-morphism.*

$$
\begin{tikzcd}
& B \arrow[dr, "i_1"] \arrow[drr, bend left, "a"] & & \\
A \arrow[ur, "p_1"] \arrow[dr, "p_2"'] \arrow[r, phantom, "(1)"] & & D \arrow[r, "u"] & X \\
& C \arrow[ur, "i_2"'] \arrow[urr, bend right, "b"'] & &
\end{tikzcd}
$$

Nested conditions in this paper are defined as application conditions for rules in [13]. Depending on the context in which a nested condition occurs, we use the terms application condition [13] and constraint [6], respectively. Furthermore, we define positive nested conditions to be used in Sec. 3, Sec. 4, and Sec. 5 for our main results.

Definition 2.4 (Nested Condition)**.** *A* nested condition *ac_P over an object P is inductively defined as follows:*

- *$true$ is a nested condition over P.*
- *For every morphism $a : P \to C$ and nested condition ac_C over C, $\exists\ (a, ac_C)$ is a nested condition over P.*
- *A nested condition can also be a Boolean formula over nested conditions. This means that also $\neg ac_P$, $\bigwedge_{i\in\mathcal{I}} ac_{P,i}$, and $\bigvee_{i\in\mathcal{I}} ac_{P,i}$ are nested conditions over P for nested conditions ac_P, $ac_{P,i}$ $(i \in \mathcal{I})$ over P for some index set $\mathcal{I}$.*

Furthermore, we distinguish the following concepts:

- *A nested condition is called* application condition *in the context of rules and match morphisms.*
- *A nested condition is called* constraint *in the context of properties of objects.*
- *A* positive nested condition *is built up only by nested conditions of the form $true$, $\exists\ (a, ac)$, $\bigwedge_{i\in\mathcal{I}} ac_{P,i}$ and $\bigvee_{i\in\mathcal{I}} ac_{P,i}$, where $\mathcal{I} \neq \emptyset$.*

An example for a nested condition and its meaning is given below.

Example 2.5 (Nested Condition)**.** *Given the nested condition ac_P from Fig. 2 where all morphisms are inclusions. Condition ac_P means that the source of every* `b`*-edge has a* `b`*-self-loop and must be followed by some* `c`*-edge such that subsequently, there is a path in the reverse direction visiting the source and target of the first* `b`*-edge with precisely one* `c`*-edge and one* `b`*-edge in an arbitrary order. We denote this nested condition by $ac_P = \exists\,(a_1, true) \wedge \exists\,(a_2, \exists\,(a_3, true) \vee \exists\,(a_4, true))$.*

We are now defining inductively whether a morphism satisfies a nested condition (see [6]).

Definition 2.6 (Satisfaction of Nested Condition)**.** *Given a nested condition ac_P over P, a morphism $p : P \to G$* satisfies *ac_P (see Fig. 1(a)), written $p \models ac_P$, if:*

- *$ac_P = true$, or*
- *$ac_P = \exists\,(a, ac_C)$ with $a : P \to C$ and there exists a morphism $q : C \to G \in \mathcal{M}$ such that $q \circ a = p$ and $q \models ac_C$, or*
- *$ac_P = \neg ac'_P$ and $p \not\models ac'_P$, or*
- *$ac_P = \bigwedge_{i \in \mathcal{I}} ac_{P,i}$ and for all $i \in \mathcal{I}$ holds $p \models ac_{P,i}$, or*
- *$ac_P = \bigvee_{i \in \mathcal{I}} ac_{P,i}$ and for some $i \in \mathcal{I}$ holds $p \models ac_{P,i}$.*

In the following we distinguish two kinds of satisfaction relations for constraints: General [6] and initial satisfaction [14]. Initial satisfaction is defined for constraints over an initial object of the base category while general satisfaction is considered for constraints over arbitrary objects. Intuitively, while general satisfaction requires that a constraint ac_P is satisfied by every $\mathcal{M}$-morphism $p : P \to G$, intial satisfaction requires just the existence of an $\mathcal{M}$-morphism $p : P \to G$ which satisfies ac_P.

$$\begin{tikzcd} P \arrow[rr, "a"] \arrow[dr, "p"'] & {} \arrow[d, phantom, "="] & C \lhd ac_C \arrow[dl, "q \models ac_C"] \\ & G & \end{tikzcd}$$

(a) Satisfaction of ac_P by morphism p

$$\begin{tikzcd} I \arrow[rr, "i_P"] \arrow[dr, "i_G"'] & {} \arrow[d, phantom, "="] & P \lhd ac_P \arrow[dl, "p \models ac_P"] \\ & G & \end{tikzcd}$$

(b) Initial satisfaction of ac_I

Figure 1: Satisfaction of nested conditions

Definition 2.7 (General Satisfaction of Constraints)**.** *Given a constraint ac_P over P. An object G generally satisfies ac_P, written $G \models ac_P$, if $\forall\ p : P \to G \in \mathcal{M}.\ p \models ac_P$ (see Fig. 1(a)).*

Definition 2.8 (Initial Satisfaction of Constraints)**.** *Given a constraint ac_I over an initial object I. An object G initially satisfies ac_I, written $G \overset{I}{\models} ac_I$, if $i_G \models ac_I$ for the initial morphism $i_G : I \to G$.*
Note, that for $ac_I = \exists\,(i_P, ac_P)$ we have

$$G \overset{I}{\models} ac_I \Leftrightarrow \exists\ p : P \to G \in \mathcal{M}.\ p \models ac_P \ \ \textit{(see Fig. 1(b))}.$$

This means that the general satisfaction corresponds more to the universal satisfaction of constraints while the initial satisfaction corresponds more to the existential satisfaction.

For positive nested conditions, we define solutions for the satisfaction problem. A solution Q (a tree of morphisms) determines which morphisms are used to fulfill the satisfaction condition.

Definition 2.9 (Solution for Satisfaction of Positive Nested Conditions)**.** *Given a positive nested condition ac_P over P and a morphism $p : P \to G$. Then Q is a solution for $p \models ac_P$ if:*

- *$ac_P = true$ and $Q = \emptyset$, or*

- $ac_P = \exists\,(a, ac_C)$ *with* $a: P \to C$ *and* $Q = (q, Q_C)$ *with* $\mathcal{M}$*-morphism* $q: C \to G$ *such that* $q \circ a = p$ *and* Q_C *is a solution for* $q \vDash ac_C$ *(see Fig. 1(a)), or*
- $ac_P = \bigwedge_{i\in\mathcal{I}} ac_{P,i}$ *and* $Q = (Q_i)_{i\in\mathcal{I}}$ *such that* Q_i *is a solution for* $p \vDash ac_{P,i}$ *for all* $i \in \mathcal{I}$*, or*
- $ac_P = \bigvee_{i\in\mathcal{I}} ac_{P,i}$ *and* $Q = (Q_i)_{i\in\mathcal{I}}$ *such that there is* $j \in \mathcal{I}$ *with solution* Q_j *for* $p \vDash ac_{P,j}$ *and for all* $k \in \mathcal{I}$ *with* $k \neq j$ *it holds that* $Q_k = \emptyset$.

The following example demonstrates the general and initial satisfaction of constraints and gives their corresponding solutions.

Example 2.10 (Satisfaction and Solution of Constraints)**.**

1. *General Satisfaction*
   Consider the graph G_A *from Fig. 2 below and the constraint* ac_P *from Ex. 2.5. There are two possible* $\mathcal{M}$*-morphisms* $p_1, p_2 : P \to G_A$*, where* p_1 *is an inclusion and* p_2 *maps* $\mathtt{b}_1$ *to* $\mathtt{b}_2$ *with the corresponding node mapping. For both matches* p_1 *and* p_2*, there is a* b*-self-loop on the image of node* 1*, a* c*-edge outgoing from the image of node* 2*, as well as the corresponding images for edges* $\mathtt{b}_2$ *and* $\mathtt{c}_2$ *in* C_3*. Thus,* G_A *generally satisfies* ac_P*.*
   A corresponding solution for $p_1 \vDash ac_P$ *is given by* $Q_{gen} = (Q_i)_{i\in\{1,2\}}$ *with* $Q_1 = (q_1, \emptyset)$ *and* $Q_2 = (q_2, (Q_j)_{j\in\{3,4\}})$*, where* $Q_3 = (q_3, \emptyset)$*,* $Q_4 = \emptyset$ *and* $q_i : C_i \to G_A$ *for* $i = 1, 2, 3$ *are inclusions.*

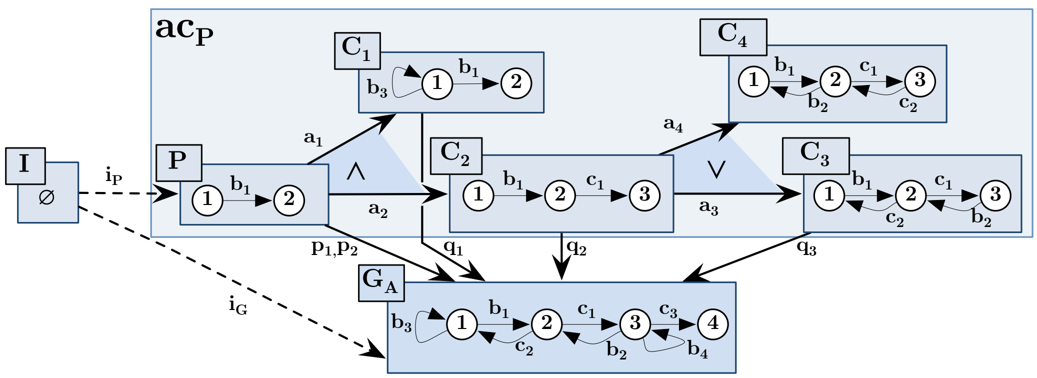


Figure 2: General and initial satisfaction of constraints

2. *Initial Satisfaction*
   Let $ac_I = \exists\,(i_P, ac_P)$ *with* i_P *as depicted in Fig. 2 and* ac_P *from Ex. 2.5. The graph* G_A *initially satisfies* ac_I *since there is* $p_1 : P \to G_A \in \mathcal{M}$ *satisfying* ac_P *as mentioned before.*
   A corresponding solution for $i_G \vDash ac_I$ *is given by* $Q_{init} = (p_1, Q_{gen})$ *with* Q_{gen} *from the example for general satisfaction.*

Remark 2.11. *A nested condition is called* typed *over a given type object, if all nested conditions in every of its nesting levels are also typed over the same type object. Furthermore, matches and corresponding solutions are required to be compatible with this type of object as well.*

3 Restriction Along Type Morphisms

In this section, we present the restriction of objects, morphisms, positive nested conditions and their solutions along type morphisms which are the basis for the amalgamation of nested conditions in Sec. 4.

General Assumption. *In this and the following sections, we consider an $\mathcal{M}$-adhesive category $(\mathbf{C},\mathcal{M})$ satisfying the horizontal VK property (see Rem. 2.2) and has effective pushouts (see Def. 2.3).*

Definition 3.1 (Restriction along Type Morphism)**.** *Given an object G_A typed over TG_A by $t_{G_A} : G_A \to TG_A$ and $t : TG_B \to TG_A \in \mathcal{M}$, then TG_B is called restriction of TG_A, G_B is a restriction of G_A, and t_{G_B} is a restriction of t_{G_A}, if (1) is a pullback. Given $a : G'_A \to G_A$, then b is a* restriction *of a along type morphism t, written $b = Restr_t(a)$, if (2) is a pullback.*

$$
\begin{tikzcd}
TG_A & G_A \arrow[l, "t_{G_A}"'] & G'_A \arrow[l, "a"'] \\
TG_B \arrow[u, "t"] \arrow[ur, phantom, "(1)"] & G_B \arrow[l, "t_{G_B}"] \arrow[u, "t_G"'] \arrow[ur, phantom, "(2)"] & G'_B \arrow[l, "b"] \arrow[u, "t'_G"']
\end{tikzcd}
$$

For positive nested conditions, we can define the restriction recursively as restriction of their components.

Definition 3.2 (Restriction of Positive Nested Conditions)**.** *Given a positive nested condition ac_{P_A} typed over TG_A and let TG_B be a restriction of it with $t : TG_B \to TG_A \in \mathcal{M}$. Then we define the restriction $ac_{P_B} = Restr_t(ac_{P_A})$ over the restriction P_B of P_A as follows:*

- *The restriction of $true$ is $true$,*
- *the restriction of $\exists\,(a, ac_{C_A})$ is given by restriction of a and ac_{C_A}, i. e., $ac_{P_B} = \exists\,(Restr_t(a), Restr_t(ac_{C_A}))$, and*
- *the restriction of a Boolean formula is given by the restrictions of its components, i. e., $Restr_t(\neg ac'_{P_A}) = \neg Restr_t(ac'_{P_A})$, $Restr_t(\bigwedge_{i\in\mathcal{I}} ac_{P_A,i}) = \bigwedge_{i\in\mathcal{I}} Restr_t(ac_{P_A,i})$, and $Restr_t(\bigvee_{i\in\mathcal{I}} ac_{P_A,i}) = \bigvee_{i\in\mathcal{I}} Restr_t(ac_{P_A,i})$.*

$$
\begin{tikzcd}
 & & P_A \arrow[dl, bend right, "a"'] \\
TG_A & C_A \lhd ac_{C_A} \arrow[l] & \\
TG_B \arrow[u, "t"] & C_B \lhd ac_{C_B} \arrow[l] \arrow[u, "t_C"'] & \\
 & & P_B \arrow[ul, bend left, "b"] \arrow[uuu, "t_P"']
\end{tikzcd}
$$

Now we extend the restriction construction to solutions of positive nested conditions and show in Fact 3.4 that a restriction of a solution is also a solution for the corresponding restricted constraint.

Definition 3.3 (Restriction of Solutions for Positive Nested Conditions)**.** *Given a positive nested condition ac_{P_A} typed over TG_A together with a restriction ac_{P_B} along $t : TG_B \to TG_A$. For a morphism $p_A : P_A \to G$ and a solution Q_A for $p_A \models ac_{P_A}$, the restriction Q_B of Q_A along t, written $Q_B = Restr_t(Q_A)$, is defined inductively as follows:*

- *If Q_A is empty then also Q_B is empty,*
- *if $ac_{P_A} = \exists\,(a : P_A \to C_A, ac_{C_A})$ and $Q_A = (q_A, Q_{CA})$, then $Q_B = (q_B, Q_{CB})$ such that q_B and Q_{CB} are restrictions of q_A respectively Q_{CA}, and*
- *if $ac_{P_A} = \bigwedge_{i\in\mathcal{I}} ac_{P_A,i}$ or $ac_{P_A} = \bigvee_{i\in\mathcal{I}} ac_{P_A,i}$, and $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$, then $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$ such that $Q_{B,i}$ is a restriction of $Q_{A,i}$ for all $i \in \mathcal{I}$.*

Fact 3.4 (Restriction of Solutions for Positive Nested Conditions)**.** *Given a positive nested condition ac_{P_A} and a match $p_A : P_A \to G_A$ over TG_A with restrictions $ac_{P_B} = Restr_t(ac_{P_A})$, $p_B = Restr_t(p_A)$ along $t : TG_B \to TG_A$. Then for a solution Q_A of $p_A \models ac_{P_A}$, there is a solution $Q_B = Restr_t(Q_A)$ for $p_B \models ac_{P_B}$.*

4 Amalgamation

The amalgamation of typed objects allows to combine objects of different types provided that they agree on a common subtype. This concept is already known in the context of different types of Petri net processes, such as open net processes [1] and algebraic high-level processes [7], which can be seen as special kinds of typed objects. In this section, we introduce a general definition for the amalgamation of typed objects. Moreover, we extend the concept to the amalgamation of positive nested conditions and their solutions.

As required for amalgamation, we discuss under which conditions morphisms can be composed via a span of restriction morphisms. Two morphisms g_B and g_C "agree" in a morphism g_D, if g_D can be constructed as a common restriction and can be used as a composition interface for g_B and g_C as in Def. 4.1.

Definition 4.1 (Agreement and Amalgamation of Typed Objects)**.** *Given a span $TG_B \xleftarrow{tg_{DB}} TG_D \xrightarrow{tg_{DC}} TG_C$, with $tg_{DB}, tg_{DC} \in \mathcal{M}$ and typed objects $G_B \overset{g_B}{\to} TG_B$, $G_C \overset{g_C}{\to} TG_C$ and $G_D \overset{g_D}{\to} TG_D$. We say g_B, g_C* agree *in g_D, if g_D is a restriction of g_B and g_C, i.e., $Restr_{tg_{DB}}(g_B) = g_D = Restr_{tg_{DC}}(g_C)$.*

Given pushout (1) below with all morphisms in $\mathcal{M}$ and typed objects g_B, g_C agreeing in g_D. A morphism $g_A : G_A \to TG_A$ is called amalgamation *of g_B and g_C over g_D, written $g_A = g_B +_{g_D} g_C$, if the outer square is a pushout and g_B, g_C are restrictions of g_A.*

$$\begin{tikzcd}
& & & G_D \arrow[dddlll] \arrow[dddrrr] \arrow[d, "g_D"] & & & \\
& & & TG_D \arrow[dll, "tg_{DB}"'] \arrow[drr, "tg_{DC}"] & & & \\
G_B \arrow[r, "g_B"'] \arrow[ddrrr] & TG_B \arrow[drr, "tg_{BA}"'] & & (1) & & TG_C \arrow[dll, "tg_{CA}"] & G_C \arrow[l, "g_C"] \arrow[ddlll] \\
& & & TG_A & & & \\
& & & G_A \arrow[u, "g_A"] & & &
\end{tikzcd}$$

Fact 4.2 is essentially based on the horizontal VK property.

Fact 4.2 (Amalgamation of Typed Objects)**.** *Given pushout (1) with all morphisms in $\mathcal{M}$ as in Def. 4.1.*

Composition. *Given g_B, g_C agreeing in g_D, then there exists a unique amalgamation $g_A = g_B +_{g_D} g_C$.*

Decomposition. *Vice versa, given $g_A : G_A \to TG_A$, there are unique restrictions g_B, g_C, and g_D of g_A such that $g_A = g_B +_{g_D} g_C$.*

Here and in the following, uniqueness means uniqueness up to isomorphism.

Proof. Given g_B, g_C agreeing in g_D, we have that the upper two trapezoids are pullbacks. Now we construct G_A as pushout over G_B and G_C via G_D, such that the outer diamond is a pushout. This leads to a unique induced morphism $g_A : G_A \to TG_A$, such that the diagram commutes and via the horizontal VK property we get that the lower two trapezoids are pullbacks and therefore $g_A = g_B +_{g_D} g_C$.

Vice versa, we can construct G_B, G_C, G_D as restrictions such that the trapezoids become pullbacks, where $g_A : G_A \to TG_A$ and TG_A, TG_B, TG_C, TG_D are given such that (1) is a pushout with $\mathcal{M}$-morphisms only. Then the horizontal VK property implies that the outer diamond is a pushout and g_A is unique because of the universal property and $g_A = g_B +_{g_D} g_C$.

The uniqueness (up to isomorphism) of the amalgamated composition and decomposition constructions follows from uniqueness of pushouts and pullpacks up to isomorphism. □

Example 4.3 (Amalgamation of Typed Objects)**.** *Figure 3 shows a pushout of type graphs TG_A, TG_B, TG_C and TG_D.*

Composition. *Consider the typed graphs G_B, G_C and G_D typed over TG_B, TG_C and TG_D, respectively. Graph G_D, containing the same nodes as G_B and G_C and no edges, is the common restriction of G_B and G_C. So, the type morphisms g_B and g_C agree in g_D, which by Fact 4.2 means that there is an amalgamation $g_A = g_B +_{g_D} g_C$. It can be obtained by computing the pushout of G_B and G_C over G_D, leading to the graph G_A that contains the* b*-edges of G_B as well as the* c*-edges of G_C. The type morphism g_A is induced by the universal property of pushouts, mapping all edges in the same way as g_B and g_C.*

Decomposition. *Vice versa, consider the graph G_A typed over TG_A. We can restrict G_A to the type graphs TG_B and TG_C, leading to typed graphs G_B and G_C, containing only the* b*- respectively* c*-edges of G_A. Restricting the graphs G_B and G_C to type graph TG_D, we get in both cases the graph G_D that contains no edges, and we have that $g_A = g_B +_{g_D} g_C$.*

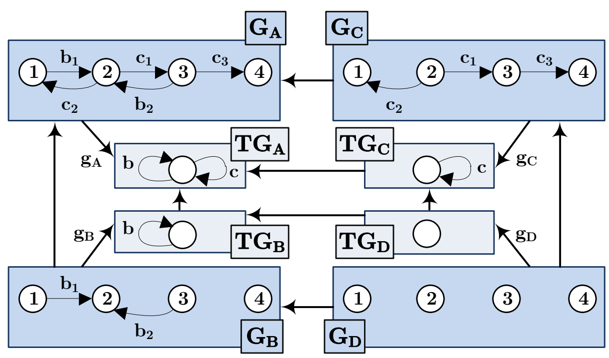


Figure 3: Amalgamation of typed graphs

We already defined the restriction of positive nested conditions (Def. 3.2) and their solutions (Def. 3.3). Now we want to consider the case that we have two conditions, which have a common restriction and can be amalgamated.

Definition 4.4 (Agreement and Amalgamation of Positive Nested Conditions)**.** *Given a pushout (1) below with all morphisms in $\mathcal{M}$. Two positive nested conditions ac_{P_B} typed over TG_B and ac_{P_C} typed over TG_C* agree *in ac_{P_D} typed over TG_D if ac_{P_D} is a restriction of ac_{P_B} and ac_{P_C}.*

Given ac_{P_B} and ac_{P_C} agreeing in ac_{P_D} then a positive nested condition ac_{P_A} typed over TG_A is called amalgamation *of ac_{P_B} and ac_{P_C} over ac_{P_D}, written $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$, if ac_{P_B} and ac_{P_C} are restrictions of ac_{P_A} and $t_{PA} = t_{PB} +_{t_{PD}} t_{PC}$. In particular, we have $true_A = true_B +_{true_D} true_C$, short $true = true +_{true} true$.*

$$\begin{tikzcd}
 & & P_D \triangleleft ac_{P_D} \arrow[dll] \arrow[drr] \arrow[d, "t_{PD}"] & & \\
 & & TG_D \arrow[dl, "tg_{DB}"'] \arrow[dr, "tg_{DC}"] & & \\
ac_{P_B} \triangleright P_B \arrow[r, "t_{PB}"'] \arrow[drr] & TG_B \arrow[dr, "tg_{BA}"'] & (1) & TG_C \arrow[dl, "tg_{CA}"] & P_C \triangleleft ac_{P_C} \arrow[l, "t_{PC}"] \arrow[dll] \\
 & & TG_A & & \\
 & & P_A \triangleleft ac_{P_A} \arrow[u, "t_{PA}"] & &
\end{tikzcd}$$

In the following Fact 4.5, we give a construction for the amalgamation of positive nested conditions and in Thm. 4.10 for the corresponding solutions.

Fact 4.5 (Amalgamation of Positive Nested Conditions)**.** *Given a pushout (1) as in Def. 4.4 with all morphisms in $\mathcal{M}$.*

Composition. *If there are positive nested conditions ac_{P_B} and ac_{P_C} typed over TG_B and TG_C, respectively, agreeing in ac_{P_D} typed over TG_D, then there exists a unique positive nested condition ac_{P_A} typed over TG_A such that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.*

Decomposition. *Vice versa, given a positive nested condition ac_{P_A} typed over TG_A, there are unique restrictions ac_{P_B}, ac_{P_C} and ac_{P_D} of ac_{P_A} such that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.*

The amalgamated composition and decomposition constructions are unique up to isomorphism.

Remark 4.6. *Given an amalgamation $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$ of positive nested conditions, we can conclude from the proof of Fact 4.5 (see App. A) that we also have corresponding amalgamations in each level of nesting.*

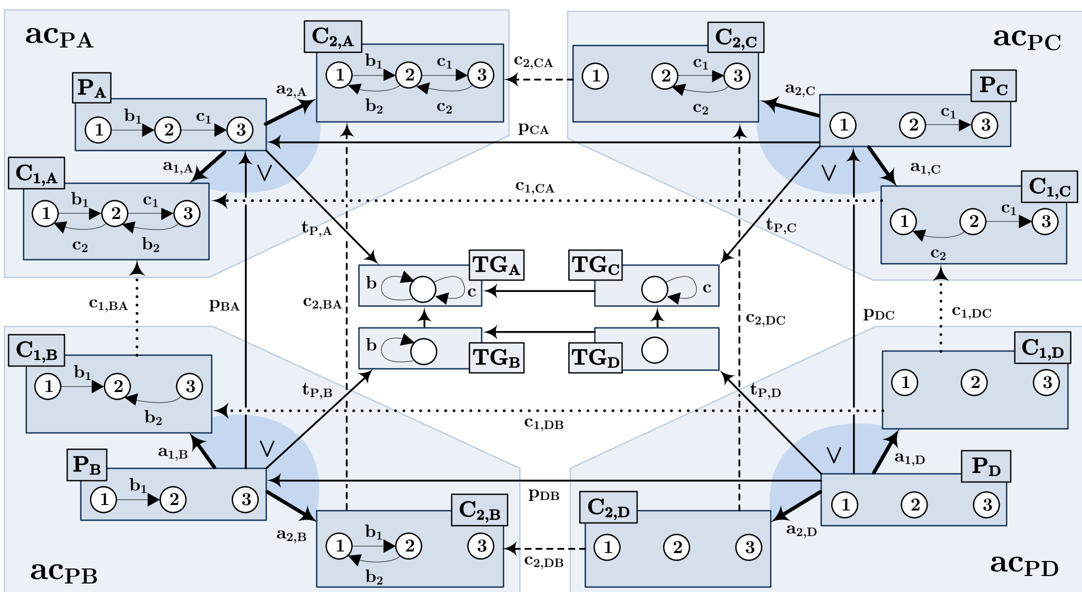


Figure 4: Amalgamation of positive nested conditions

Example 4.7 (Amalgamation of Positive Nested Conditions)**.** *Figure 4 shows a pushout of typed graphs TG_A, TG_B, TG_C and TG_D, and four positive nested conditions ac_{P_A}, ac_{P_B}, ac_{P_C} and ac_{P_D} typed over TG_A, TG_B, TG_C and TG_D, respectively. For simplicity, the figure contains only the type morphisms of the Ps, but there are also corresponding type morphisms for the Cs, mapping all* b*-edges to* b *and all* c*-edges to* c*. There is $ac_{P_A} = \bigvee_{i\in\{1,2\}} ac_{C_{i,A}}$ with $ac_{C_{i,A}} = \exists\,(a_{i,A}, true)$ for $i = 1,2$, and ac_{P_B}, ac_{P_C} and ac_{P_D} have a similar structure.*

Composition. *We have that t_{P_D} is a common restriction of t_{P_B} and t_{P_C}, and also that $a_{i,D}$ is a common restriction of $a_{i,B}$ and $a_{i,C}$ for $i = 1,2$. Thus, ac_{P_D} is a common restriction of ac_{P_B} and ac_{P_C} which means that ac_{P_B} and ac_{P_C} agree in ac_{P_D}. So by Fact 4.5 there exists an amalgamation $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$, and according to Rem. 4.6 it can be obtained as amalgamation of its components.*

This means that we have an amalgamation $t_{P_A} = t_{P_B} +_{t_{P_D}} t_{P_C}$ with pushout of the Ps as shown in Fig. 4, as well as amalgamations of the corresponding type morphisms of the Cs, leading to the pushouts depicted in Fig. 4 by dotted arrows for the C_1s and by dashed arrows for the C_2s. The morphisms $a_{1,A}$ and $a_{2,A}$ are obtained by the universal property of pushouts.

Decomposition. *The other way around, considering the condition ac_{P_A}, we can construct the restrictions ac_{P_B} and ac_{P_C} by deleting the $\mathfrak{c}$- respectively $\mathfrak{b}$-edges. Then, restricting ac_{P_B} and ac_{P_C} to TG_D by deleting all remaining edges, we obtain the same condition ac_{P_D} such that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.*

In order to answer the question, under which conditions such amalgamated positive nested conditions are satisfied, we need to define an amalgamation of their solutions. Afterwards, we show in the proof of Thm. 4.10 that a composition of two solutions via an interface leads to a unique amalgamated solution and that a given solution for an amalgamated positive nested condition is the amalgamation of its unique restrictions.

Definition 4.8 (Agreement and Amalgamation of Solutions for Positive Nested Conditions)**.** *Given pushout (1) below with all morphisms in $\mathcal{M}$, an amalgamation of typed objects $g_A = g_B +_{g_D} g_C$, and an amalgamation of positive nested conditions $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$ with corresponding matches $p_A = p_B +_{p_D} p_C$.*

1. *Two solutions Q_B for $p_B \models ac_{P_B}$ and Q_C for $p_C \models ac_{P_C}$* agree *in a solution Q_D for $p_D \models ac_{P_D}$, if Q_D is a restriction of Q_B and Q_C.*
2. *Given solutions Q_B for $p_B \models ac_{P_B}$ and Q_C for $p_C \models ac_{P_C}$ agreeing in a solution Q_D for $p_D \models ac_{P_D}$, then a solution Q_A for $p_A \models ac_{P_A}$ is called* amalgamation *of Q_B and Q_C over Q_D, written $Q_A = Q_B +_{Q_D} Q_C$, if Q_B and Q_C are restrictions of Q_A.*

$$
\begin{tikzcd}
ac_{P_A} \triangleright P_A \arrow[dr, "p_A"] & & & & & P_C \triangleleft ac_{P_C} \arrow[lllll, "p_{CA}"'] \arrow[dl, "p_C"'] \\
& G_A \arrow[dr, "g_A"] & & & G_C \arrow[lll, "g_{CA}"'] \arrow[dl, "g_C"'] & \\
& & TG_A & TG_C \arrow[l, "tg_{CA}"'] & & \\
& & TG_B \arrow[u, "tg_{BA}"] \arrow[ur, phantom, "(1)"] & TG_D \arrow[l, "tg_{DB}"] \arrow[u, "tg_{DC}"'] & & \\
& G_B \arrow[uuu, "g_{BA}"] \arrow[ur, "g_B"] & & & G_D \arrow[lll, "g_{DB}"] \arrow[ul, "g_D"'] \arrow[uuu, "g_{DC}"'] & \\
ac_{P_B} \triangleright P_B \arrow[uuuuu, "p_{BA}"] \arrow[ur, "p_B"] & & & & & P_D \triangleleft ac_{P_D} \arrow[lllll, "p_{DB}"] \arrow[ul, "p_D"'] \arrow[uuuuu, "p_{DC}"']
\end{tikzcd}
$$

Remark 4.9. *Note that by assumption $g_A = g_B +_{g_D} g_C$ in the definition above we already have a pushout over the Gs, and by $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$ we also have a pushout over the Ps.*

Theorem 4.10 (Amalgamation of Solutions for Positive Nested Conditions)**.** *Given pushout (1) as in Def. 4.8 with all morphisms in $\mathcal{M}$, an amalgamation of typed objects $g_A = g_B +_{g_D} g_C$, and an amalgamation of positive nested conditions $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$ with corresponding matches $p_A = p_B +_{p_D} p_C$.*

Composition. *Given solutions Q_B for $p_B \models ac_{P_B}$ and Q_C for $p_C \models ac_{P_C}$ agreeing in a solution Q_D for $p_D \models ac_{P_D}$, then there is a solution Q_A for $p_A \models ac_{P_A}$ constructed as amalgamation $Q_A = Q_B +_{Q_D} Q_C$.*

Decomposition. *Given a solution Q_A for $p_A \models ac_{P_A}$, then there are solutions Q_B, Q_C and Q_D for $p_B \models ac_{P_B}$, $p_C \models ac_{P_C}$ and $p_D \models ac_{P_D}$, respectively, which are constructed as restrictions Q_B, Q_C and Q_D of Q_A such that $Q_A = Q_B +_{Q_D} Q_C$.*

The amalgamated composition and decomposition constructions are unique up to isomorphism.

Remark 4.11. *From the proof of Thm. 4.10 (see App. A) we can conclude that for a given amalgamation of solutions $Q_A = Q_B +_{Q_D} Q_C$, we also have corresponding amalgamations of its components.*

5 Compatibility of Initial Satisfaction with Restriction and Amalgamation

In this section we present our main result showing compatibility of initial satisfaction with amalgamation (Thm. 5.1) and restriction (Cor. 5.2) which are based on the amalgamation of solutions for positive nested conditions (Thm. 4.10). This main result allows to conclude the satisfaction of a constraint for a composed object from the satisfaction of the corresponding restricted constraints for the component objects. It is valid for initial satisfaction, but not for general satisfaction.

Theorem 5.1 (Compatibility of Initial Satisfaction with Amalgamation)**.** *Given pushout (1) below with all morphisms in $\mathcal{M}$, an amalgamation of typed objects $g_A = g_B +_{g_D} g_C$, and an amalgamation of positive constraints $ac_A = ac_B +_{ac_D} ac_C$. Then we have:*

- ***Decomposition.*** *Given a solution Q_A for $G_A \overset{I}{\models} ac_A$, then there are solutions Q_B for $G_B \overset{I}{\models} ac_B$, Q_C for $G_C \overset{I}{\models} ac_C$ and Q_D for $G_D \overset{I}{\models} ac_D$ such that $Q_A = Q_B +_{Q_D} Q_C$.*
- ***Composition.*** *Vice versa, given solutions Q_B for $G_B \overset{I}{\models} ac_B$ and Q_C for $G_C \overset{I}{\models} ac_C$ agreeing in a solution Q_D for $G_D \overset{I}{\models} ac_D$, then there exists a solution Q_A for $G_A \overset{I}{\models} ac_A$ such that $Q_A = Q_B +_{Q_D} Q_C$.*

$$\begin{tikzcd}
ac_A \triangleright I \arrow[dr, "i_{G_A}"] & & & & & I \triangleleft ac_C \arrow[lllll, "id_I"'] \arrow[dl, "i_{G_C}"'] \\
& G_A \arrow[dr, "g_A"'] & & & G_C \arrow[lll, "g_{CA}"'] \arrow[dl, "g_C"] & \\
& & TG_A & TG_C \arrow[l, "tg_{CA}"'] & & \\
& & TG_B \arrow[u, "tg_{BA}"] \arrow[ur, phantom, "(1)"] & TG_D \arrow[l, "tg_{DB}"] \arrow[u, "tg_{DC}"'] & & \\
& G_B \arrow[uuu, "g_{BA}"'] \arrow[ur, "g_B"] & & & G_D \arrow[lll, "g_{DB}"] \arrow[uuu, "g_{DC}"] \arrow[ul, "g_D"'] & \\
ac_B \triangleright I \arrow[uuuuu, "id_I"] \arrow[ur, "i_{G_B}"'] & & & & & I \triangleleft ac_D \arrow[lllll, "id_I"] \arrow[uuuuu, "id_I"'] \arrow[ul, "i_{G_D}"]
\end{tikzcd}$$

Proof.

- **Decomposition.** By Def. 2.8 a solution Q_A for $G_A \overset{I}{\models} ac_A$ is also a solution for $i_{G_A} \models ac_A$, where i_{G_A} is the unique morphism $i_{G_A} : I \to G_A$. Moreover, due to amalgamation $g_A = g_B +_{g_D} g_C$ the inner trapezoids in the diagram above are pullbacks. So by closure of $\mathcal{M}$ under pullbacks we have that $g_{BA}, g_{CA}, g_{DB}, g_{DC} \in \mathcal{M}$ which means that they are monomorphisms. Therefore, the outer trapezoids become pullbacks by standard category theory, which means that $i_{G_B} : I \to G_B$ is a restriction of i_{G_A}, $i_{G_C} : I \to G_C$ is a restriction of i_{G_A}, and $i_{G_D} : I \to G_D$ is a restriction of i_{G_B} as well as of i_{G_C}.

  Furthermore, the outer square in the diagram is a pushout, implying that we have an amalgamation $i_{G_A} = i_{G_B} +_{i_{G_D}} i_{G_C}$. Thus, using Thm. 4.10 we obtain solutions Q_B for $i_{G_B} \models ac_B$, Q_C for $i_{G_C} \models ac_C$ and Q_D for $i_{G_D} \models ac_D$ such that $Q_A = Q_B +_{Q_D} Q_C$, and by Def. 2.8 Q_B, Q_C and Q_D are solutions for $G_B \overset{I}{\models} ac_B$, $G_C \overset{I}{\models} ac_C$ and $G_D \overset{I}{\models} ac_D$, respectively.

- **Composition.** Now, given solutions Q_B, Q_C and Q_D for $G_B \overset{I}{\models} ac_B$, $G_C \overset{I}{\models} ac_C$ and $G_D \overset{I}{\models} ac_D$, respectively. Then by Def. 2.8 we have that Q_B, Q_C and Q_D are solutions for $i_{G_B} \models ac_B$, $i_{G_C} \models ac_C$ and $i_{G_D} \models ac_D$, respectively. As shown in item 1, there is $i_{G_A} = i_{G_B} +_{i_{G_D}} i_{G_C}$ and therefore, since Q_B and Q_C agree in Q_D, by Thm. 4.10 we obtain a solution Q_A for $i_{G_A} \models ac_A$ such that $Q_A = Q_B +_{Q_D} Q_C$. Finally, Def. 2.8 implies that Q_A is a solution for $G_A \overset{I}{\models} ac_A$.

□

Corollary 5.2 (Compatibility of Initial Satisfaction with Restriction)**.** *Given type restriction $t : TG_B \to TG_A \in \mathcal{M}$, object G_A typed over TG_A with restriction G_B, and a positive constraint ac_A over initial object I typed over TG_A with restriction ac_B. Then $G_A \overset{I}{\models} ac_A$ implies $G_B \overset{I}{\models} ac_B$. Moreover, if Q_A is a solution for $G_A \overset{I}{\models} ac_A$ then $Q_B = Restr_t(Q_A)$ is a solution for $G_B \overset{I}{\models} ac_B$.*

Proof. Consider the diagram in Thm. 5.1 with $G_C = G_A$, $G_D = G_B$, $ac_C = ac_A$ and $ac_D = ac_B$. Then by standard category theory we have that all rectangles in the diagram are pushouts and the trapezoids are pullbacks. Thus, we have $g_A = g_B +_{g_B} g_A$ and, analogously, $ac_A = ac_B +_{ac_B} ac_A$ with corresponding matches $i_{G_A} = i_{G_B} +_{i_{G_B}} i_{G_A}$. So, given a solution Q_A for $G_A \overset{I}{\models} ac_A$, by item 1 of Thm. 5.1 there is a solution Q_B for $G_B \overset{I}{\models} ac_B$ with $Q_A = Q_B +_{Q_B} Q_A$ such that by Def. 4.4 Q_B is a restriction of Q_A. □

Example 5.3 (Compatibility of Initial Satisfaction with Amalgamation)**.** *Figure 5 shows the amalgamation of typed graphs $g_A = g_B +_{g_D} g_C$ from Ex. 4.3 and an amalgamation of positive nested conditions $ac_A = ac_B +_{ac_D} ac_C$. Note that we have $ac_A = \exists\, (i_{P_A}, ac_{P_A})$ and ac_B, ac_C and ac_D with similar structure, where the amalgamation $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$ is presented in Ex. 4.7.*

Composition. *For $G_B \overset{I}{\models} ac_B$ we have the solution $Q_B = (q_B, (Q_{1,B}, Q_{2,B}))$ with $Q_{1,B} = (q_{1,B}, \emptyset)$ and $Q_{2,B} = \emptyset$, where q_B and $q_{1,B}$ are inclusions. Moreover, we have similar solutions Q_C for $G_C \overset{I}{\models} ac_C$ and Q_D for $G_D \overset{I}{\models} ac_D$. According to Rem. 4.11, the amalgamation $Q_A = Q_B +_{Q_D} Q_C$ can be constructed by amalgamation of the components.*

First, we explain in detail the amalgamation $q_{1,A} = q_{1,B} +_{q_{1,D}} q_{1,C}$. Note that the graphs G_A, G_B, G_C and G_D can be considered as type graphs such that e. g. $C_{1,D}$ is typed over G_D by $q_{1,D}$. So, since $q_{1,D}$ is a common restriction of $q_{1,B}$ and $q_{1,C}$, we have that $q_{1,B}$ and $q_{1,C}$ agree in $q_{1,D}$. This means that there is an amalgamation of typed objects $q_{1,A} = q_{1,B} +_{q_{1,D}} q_{1,C}$, where the inclusion $q_{1,A}$ maps all nodes and edges in the same way as $q_{1,B}$ and $q_{1,C}$.

Moreover, for the empty solutions we have an empty solution as amalgamation, and thus, we have amalgamations of solutions $Q_{1,A} = Q_{1,B} +_{Q_{1,D}} Q_{1,C} = (q_{1,A}, \emptyset)$ and $Q_{2,A} = Q_{2,B} +_{Q_{2,D}} Q_{2,C} = \emptyset$. The amalgamation $q_A = q_B +_{q_D} q_C$ can be obtained analogously as described for $q_{1,A}$, and hence, we have $Q_A = Q_B +_{Q_D} Q_C = (q_A, (Q_{1,A}, Q_{2,A}))$, which is a solution for $G_A \overset{I}{\models} ac_A$.

Decomposition. *For $G_A \overset{I}{\models} ac_A$ we have a solution $Q_A = (q_A, (Q_{1,A}, Q_{2,A}))$ with $Q_{1,A} = (q_{1,A}, \emptyset)$ and $Q_{2,A} = \emptyset$ where q_A and $q_{1,A}$ are inclusions. The restrictions Q_B, Q_C and Q_D of Q_A are given by restrictions of the components. By computing the restrictions $q_{1,B}$, $q_{1,C}$ and $q_{1,D}$ of $q_{1,A}$, and similar the restrictions of q_A and $\emptyset$, we get as result again the solutions Q_B for $G_B \overset{I}{\models} ac_B$, Q_C for $G_C \overset{I}{\models} ac_C$, and Q_D for $G_D \overset{I}{\models} ac_D$ as described in the composition case above.*

From Cor. 5.2, we know that initial satisfaction is compatible with restriction of typed objects and constraints. In contrast, general satisfaction and restriction are not compatible in general. As the following example illustrates, it is possible that a typed object generally satisfies a constraint while the same does not hold for their restrictions.

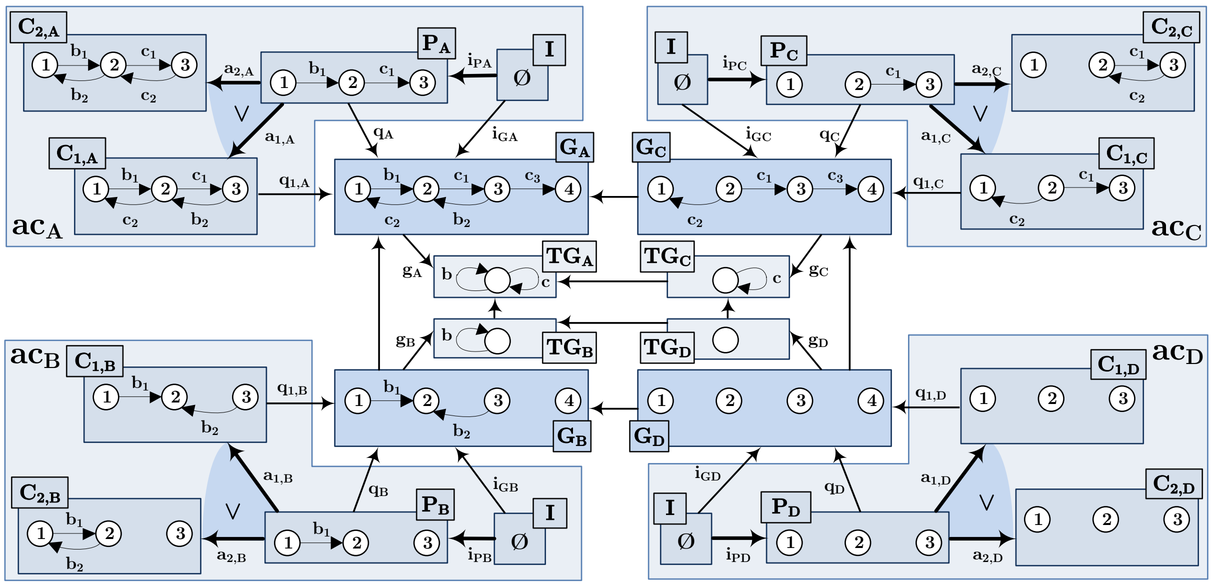


Figure 5: Amalgamation of solutions for initial satisfaction

Example 5.4 (Restriction of General Satisfaction Fails in General)**.** *Figure 6 shows a restriction G_B of the typed graph G_A and a restriction ac_{P_B} of constraint ac_{P_A}. There are two possible matches $p_{1,A}, p_{2,A} : P_A \to G_A \in \mathcal{M}$ where $p_{1,A}$ is an inclusion and $p_{2,A}$ maps $\mathtt{b}_1$ to $\mathtt{b}_2$ and $\mathtt{c}_1$ to $\mathtt{c}_2$. Since for each of the matches the graph G_A contains the required edges in the inverse direction, both of the matches satisfy ac_{P_A}. For $p_{i,A}$ we have $q_{i,A}$ with $q_{i,A} \circ a_A = p_{i,A}$ for $i = 1,2$. Thus, we have that $G_A \models ac_{P_A}$.*

For the constraint ac_{P_B} there is a match $p_B : P_B \to G_B \in \mathcal{M}$ mapping edge $\mathtt{b}_1$ identically and node 3 to node 4. We have that $p_B \not\models ac_{P_B}$ because there is no edge from node 4 to node 2 in G_B, which means that $G_B \not\models ac_{P_B}$. This is due to the fact that there is no match $p_A : P_A \to G_A \in \mathcal{M}$ such that p_B is the restriction of p_A.

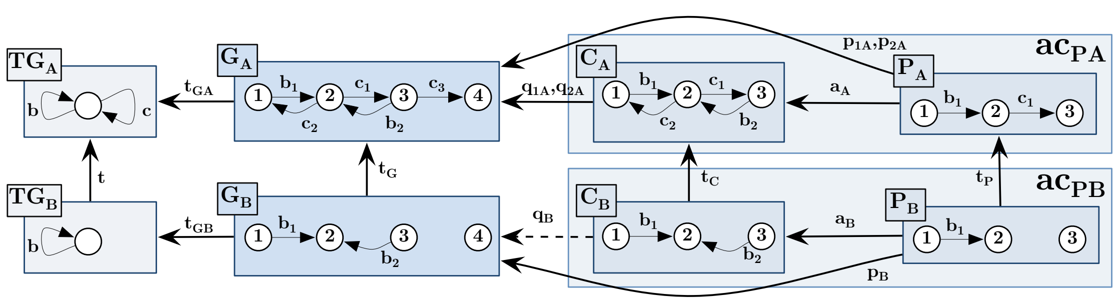


Figure 6: Counterexample for restriction of general satisfaction

6 Related Work

The framework of $\mathcal{M}$-adhesive categories [8] generalizes various kinds of categories for high level replacement systems, e.g. adhesive [17], quasi-adhesive [18], partial VK square adhesive [15], and weak-adhesive categories [6]. Therefore, the results of this paper are applicable to all of them, where the category of typed attributed graphs is a prominent example.

The concepts of nested graph conditions [13] and first-order graph formulas [4] are shown to be expressively equivalent in [14] using the translation between first-order logic and predicates on edge-labeled graphs without parallel edges [20].

Multi-view modelling is an important concept in software engineering. Several approaches have been studied and used, e.g. focussing on aspect oriented techniques [12]. In this line, graph transformation (GT) approaches have been extended to support view concepts based on the integration of type graphs. For this purpose, the concept of restriction along type morphisms has been studied and used intensively [11, 5] including GT systems using the concept of inheritance and views [5, 16]. Instead of restriction of constraints considered in this paper, only more restrictive forward translations of view constraints have been studied in [5] for the case of atomic constraints with general satisfaction leading to a result similar to Thm. 5.1. The notions of initial and general satisfaction for nested conditions can be transformed one into the other [14], but this transformation uses the Boolean operator negation that is not present in positive constraints, for which, however, our main result on the compatibility of restriction and initial satisfaction holds. Moreover, we have shown by counterexample that general satisfaction is not compatible with restriction in general, even if only positive constraints are considered.

7 Conclusion

Nested application conditions for rules and constraints for graphs and more general models have been studied already in the framework of $\mathcal{M}$-adhesive transformation systems [6, 9]. The new contribution of this paper is to study compatibility of satisfaction with restriction and amalgamation. This is important for large typed systems respectively objects, which can be decomposed by restriction and composed by amalgamation. The main result in this paper shows that initial satisfaction of positive constraints is compatible with restriction and amalgamation (Thm. 5.1 and Cor. 5.2). The amalgamation construction is based on the horizontal van Kampen (VK) property, which is required in addition to the vertical VK property of $\mathcal{M}$-adhesive categories. To our best knowledge, this is the most interesting result for $\mathcal{M}$-adhesive transformation systems which is based on the horizontal VK property. Note that the main result is not valid for general satisfaction of positive constraints nor for initial satisfaction of general constraints. For future work, it is important to obtain weaker versions of the main result, which are valid for general satisfaction and constraints, respectively.

References

[1] P. Baldan, A. Corradini, H. Ehrig & R. Heckel (2001): *Compositional Modeling of Reactive Systems Using Open Nets*. In K. G. Larsen & M. Nielse, editors: *Proc. of CONCUR 2001*, *LNCS* 2154, Springer, pp. 502–518, doi:10.1007/3-540-44685-0_34.

[2] E. Biermann, H. Ehrig, C. Ermel, K. Hoffmann & T. Modica (2009): *Modeling Multicasting in Dynamic Communication-based Systems by Reconfigurable High-level Petri Nets*. In: *Proc. of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2009)*, IEEE, pp. 47–50, doi:10.1109/VLHCC.2009.5295303.

[3] B. Braatz, H. Ehrig, K. Gabriel & U. Golas (2010): *Finitary M-Adhesive Categories*. In H. Ehrig, A. Rensink, G. Rozenberg & A. Schürr, editors: *Proc. ICGT 2010*, *LNCS* 6372, Springer, pp. 234–249, doi:10.1007/978-3-642-15928-2_16.

[4] B. Courcelle (1997): *The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic*. In Grzegorz Rozenberg, editor: *Handbook of Graph Grammars*, World Scientific, pp. 313–400.

[5] H. Ehrig, K. Ehrig, C. Ermel & U. Prange (2010): *Consistent Integration of Models based on Views of Meta Models*. *Formal Aspects of Computing* 22 (3), pp. 327–345, doi:10.1007/s00165-009-0127-6.

[6] H. Ehrig, K. Ehrig, U. Prange & G. Taentzer (2006): *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theor. Comp. Science, Springer.

[7] H. Ehrig & K. Gabriel (2011): *Transformation of Algebraic High-Level Nets and Amalgamation of Processes with Applications to Communication Platforms*. *Festschrift in Honour of Manfred Broy's 60th Birthday. International Journal of Software and Informatics* 5(1-2,Part1).

[8] H. Ehrig, U. Golas & F. Hermann (2010): *Categorical Frameworks for Graph Transformation and HLR Systems based on the DPO Approach*. *Bulletin of the EATCS* 102, pp. 111–121.

[9] H. Ehrig, A. Habel, J. Padberg & U. Prange (2006): *Adhesive High-Level Replacement Systems: A New Categorical Framework for Graph Transformation*. *Fundamenta Informaticae* 74(1), pp. 1–29.

[10] H. Ehrig, K. Hoffmann, J. Padberg, C. Ermel, U. Prange, E. Biermann & T. Modica (2008): *Petri Net Transformations*. In: *Petri Net Theory and Applications*, I-Tech Education and Publication, pp. 1–16, doi:10.5772/5310.

[11] G. Engels, R. Heckel, G. Taentzer & H. Ehrig (1997): *A Combined Reference Model- and View-Based Approach to System Specification*. *International Journal of Software Engineering and Knowledge Engineering* 7(4), pp. 457–477.

[12] R. B. France, I. Ray, G. Georg & S. Ghosh (2004): *Aspect-oriented approach to early design modelling*. *IEE Proceedings - Software* 151(4), pp. 173–186, doi:10.1049/ip-sen:20040920.

[13] A. Habel & K.-H. Pennemann (2005): *Nested constraints and application conditions for high-level structures*. In H.-J. Kreowski, U. Montanari, F. Orejas, G. Rozenberg & G. Taentzer, editors: *Formal Methods in Software and Systems Modeling*, *LNCS* 3393, Springer, pp. 294–308, doi:10.1007/978-3-540-31847-7_17.

[14] A. Habel & K.-H. Pennemann (2009): *Correctness of high-level transformation systems relative to nested conditions*. *Mathematical Structures in Computer Science* 19, pp. 1–52, doi:10.1017/S0960129508007202.

[15] T. Heindel (2010): *Hereditary Pushouts Reconsidered*. In H. Ehrig, A. Rensink, G. Rozenberg & A. Schürr, editors: *Proc. ICGT 2010*, *LNCS* 6372, Springer, pp. 250–265, doi:10.1007/978-3-642-15928-2_17.

[16] S. Jurack & G. Taentzer (2010): *A Component Concept for Typed Graphs with Inheritance and Containment Structures*. In H. Ehrig, A. Rensink, G. Rozenberg & A. Schürr, editors: *Proc. ICGT 2010*, *LNCS* 6372, Springer, pp. 187–202, doi:10.1007/978-3-642-15928-2_13.

[17] S. Lack & P. Sobociński (2004): *Adhesive Categories*. In Igor Walukiewicz, editor: *Proc. FOSSACS 2004*, *LNCS* 2987, Springer, pp. 273–288, doi:10.1007/978-3-540-24727-2_20.

[18] S. Lack & P. Sobociński (2005): *Adhesive and quasiadhesive categories*. *ITA* 39(3), pp. 511–545, doi:10.1051/ita:2005028.

[19] J. de Lara, R. Bardohl, H. Ehrig, K. Ehrig, U. Prange & G. Taentzer (2007): *Attributed Graph Transformation with Node Type Inheritance*. *TCS* 376(3), pp. 139–163, doi:10.1016/j.tcs.2007.02.001.

[20] A. Rensink (2004): *Representing First-Order Logic Using Graphs*. In H. Ehrig, G. Engels, F. Parisi-Presicce & G. Rozenberg, editors: *Proc. ICGT 2004*, *LNCS* 3256, Springer, pp. 319–335, doi:10.1007/978-3-540-30203-2_23.

[21] H. Schölzel, H. Ehrig, M. Maximova, K. Gabriel & F. Hermann (2012): *Satisfaction, Restriction and Amalgamation of Constraints in the Framework of M-Adhesive Categories: Extended Version*. Technical Report 2012/1, TU Berlin, Fak. IV. Available at `http://www.eecs.tu-berlin.de/menue/forschung/forschungsberichte/2012`.

A Remaining Proofs

In this appendix, we give the proofs for Fact 3.4, Fact 4.5 and Thm. 4.10.

Fact 3.4 (Restriction of Solutions for Positive Nested Conditions)**.** *Given a positive nested condition ac_{P_A} and a match $p_A : P_A \to G_A$ over TG_A with restrictions $ac_{P_B} = Restr_t(ac_{P_A})$, $p_B = Restr_t(p_A)$ along $t : TG_B \to TG_A$. Then for a solution Q_A of $p_A \models ac_{P_A}$ there is a solution $Q_B = Restr_t(Q_A)$ for $p_B \models ac_{P_B}$.*

Proof.

- For $ac_{P_A} = true$ the implication is trivial, because Q_A is empty which means that also Q_B is empty and thus a solution for $p_B \models ac_{P_B}$ is empty, because ac_{P_B} is also $true$.
- For $ac_{P_A} = \exists\,(a, ac_{C_A})$ we have that $Q_A = (q_A, Q_{CA})$ such that $q_A : C_A \to G_A \in \mathcal{M}$ with $q_A \circ a = p_A$ and Q_{CA} is a solution for $q_A \models ac_{C_A}$. Then by $q_B = Restr_t(q_A) : C_B \to G_B$, we have $q_B \in \mathcal{M}$ and we also have $t_G : G_B \to G_A \in \mathcal{M}$, because $t \in \mathcal{M}$ (see Fig. 7). So for $ac_{P_B} = \exists\,(b, ac_{C_B})$ we have

  $$t_G \circ q_B \circ b = q_A \circ t_C \circ b = q_A \circ a \circ t_P = p_A \circ t_P = t_G \circ p_B,$$

  which by monomorphism t_G implies $q_B \circ b = p_B$.

  Moreover, the fact that Q_{CA} is a solution for $q_A \models ac_{C_A}$ implies that $Q_{CB} = Restr_t(Q_{CA})$ is a solution for $q_B \models ac_{C_B}$ by induction hypothesis and hence the restriction $Q_B = (q_B, Q_{CB})$ of Q_A is a solution for $p_B \models ac_{P_B}$.
- Now, for $ac_{P_A} = \bigwedge_{i\in\mathcal{I}} ac_{P_A,i}$ we have $ac_{P_B} = \bigwedge_{i\in\mathcal{I}} Restr_t(ac_{P_A,i})$. By the fact that Q_A is a solution for $p_A \models ac_{P_A}$, we have that $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ such that $Q_{A,i}$ is a solution for $p_A \models ac_{A,i}$ for all $i \in \mathcal{I}$. Thus, by induction hypothesis, we have restrictions $Q_{B,i} = Restr_t(Q_{A,i})$ that are solutions for $p_B \models Restr_t(ac_{P_A,i})$ for all $i \in \mathcal{I}$. Hence, the restriction $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$ of Q_A is a solution for $p_B \models ac_{P_B}$.
- Finally, for $ac_{P_A} = \bigvee_{i\in\mathcal{I}} ac_{P_A,i}$ we have $ac_{P_B} = \bigvee_{i\in\mathcal{I}} Restr_t(ac_{P_A,i})$. By the fact that Q_A is a solution for $p_A \models ac_{P_A}$ we have that $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ such that for one $j \in \mathcal{I}$ there is a solution $Q_{A,j}$ for $p_A \models ac_{A,j}$ and for all $k \neq j$ we have that $Q_{A,k} = \emptyset$. Thus, by induction hypothesis, the restriction $Q_{B,j}$ of $Q_{A,j}$ is a solution for $p_B \models Restr_t(ac_{P_A,j})$. Hence, we also have that the restriction $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$ is a solution for $p_B \models ac_{P_B}$ with $Q_{B,k} = \emptyset$ for $k \neq j$.

□

Fact 4.5 (Amalgamation of Positive Nested Conditions)**.** *Given a pushout (1) as in Def. 4.4 with all morphisms in $\mathcal{M}$.*

$$\begin{tikzcd}
 & & & P_A \arrow[dl, "a"'] \arrow[dll, bend right, "p_A"'] \\
TG_A & G_A \arrow[l, "t_{G_A}"'] & C_A \triangleleft ac_{C_A} \arrow[l, "q_A"'] & \\
TG_B \arrow[u, "t"] & G_B \arrow[l, "t_{G_B}"'] \arrow[u, "t_G"'] & C_B \triangleleft ac_{C_B} \arrow[l, "q_B"'] \arrow[u, "t_C"'] & \\
 & & & P_B \arrow[ul, "b"] \arrow[ull, bend left, "p_B"] \arrow[uuu, "t_P"']
\end{tikzcd}$$

Figure 7: Restriction of solution q_A for $p_A \models \exists\,(a, ac_{C_A})$

Composition. *If there are positive nested conditions ac_{P_B} and ac_{P_C} typed over TG_B and TG_C, respectively, agreeing in ac_{P_D} typed over TG_D then there exists a unique positive nested condition ac_{P_A} typed over TG_A such that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.*

Decomposition. *Vice versa, given a positive nested condition ac_{P_A} typed over TG_A, there are unique restrictions ac_{P_B}, ac_{P_C} and ac_{P_D} of ac_{P_A} such that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.*

The amalgamated composition and decomposition constructions are unique up to isomorphism.

Proof.

Composition. We perform an induction over the structure of ac_{P_D}:

- $ac_{P_D} = true$.
  Then we also have $ac_{P_B} = true$ and $ac_{P_C} = true$, and the amalgamation ac_{P_A} is trivially given by $ac_{P_A} = true$.
- $ac_{P_D} = \exists\,(d, ac_{C_D})$ with $d : P_D \to C_D$.
  The assumption that ac_{P_B} and ac_{P_C} agree in ac_{P_D} means that ac_{P_D} is a restriction of ac_{P_B} and ac_{P_C} and thus, by Def. 3.2, we have that $ac_{P_B} = \exists\,(b, ac_{C_B})$ with $b : P_B \to C_B$, $ac_{P_C} = \exists\,(c, ac_{C_C})$ with $c : P_C \to C_C$, d is a restriction of b and c, and ac_{C_D} is a restriction of ac_{C_B} and ac_{C_C}. This in turn means that ac_{C_B} and ac_{C_C} agree in ac_{C_D} according to Def. 4.4. So, by induction hypothesis, we obtain an amalgamation $ac_{C_A} = ac_{C_B} +_{ac_{C_D}} ac_{C_C}$, which implies that $t_{CA} = t_{CB} +_{t_{CD}} t_{CC}$, i. e., diagrams (2)-(5) below are pullbacks. By closure of $\mathcal{M}$ under pullbacks, we obtain from $tg_{BA}, tg_{CA} \in \mathcal{M}$ that also $c_{BA}, c_{CA} \in \mathcal{M}$.
  Moreover, the fact that d is a restriction of b and c means that (6)+(2) and (7)+(3) are pullbacks, which by pullback decomposition implies that (6) and (7) are pullbacks. Note that b, c and d can be considered as typed over C_B, C_C and C_D, respectively. So, according to Def. 4.1, we obtain that b and c agree in d with respect to the pushout of the Cs, leading to an amalgamation $a = b +_d c : P_A \to C_A$ with pullbacks (8) and (9) by Fact 4.2. Hence, $ac_{P_A} = \exists\,(a, ac_{C_A})$ is the required amalgamation.

$$
\begin{tikzcd}
ac_{P_A} \triangleright P_A \arrow[dr, "a"] & & & & & P_C \lhd ac_{P_C} \arrow[lllll, "p_{CA}"'] \arrow[dl, "c"'] \\
& \overset{\substack{ac_{C_A} \\ \triangledown}}{C_A} \arrow[dr, "t_{CA}"] \arrow[rrr, phantom, "(8)", shift left=3] \arrow[rrr, phantom, "(4)"', shift right=3] & & & \overset{\substack{ac_{C_C} \\ \triangledown}}{C_C} \arrow[lll, "c_{CA}"'] \arrow[dl, "t_{CC}"'] & \\
\arrow[dr, phantom, "(9)"] & \arrow[dr, phantom, "(5)"] & TG_A \arrow[dr, phantom, "(1)"] & TG_C \arrow[l, "tg_{CA}"'] \arrow[dr, phantom, "(3)"] & \arrow[dr, phantom, "(7)"] & \\
& & TG_B \arrow[u, "tg_{BA}"] & TG_D \arrow[l, "tg_{DB}"'] \arrow[u, "tg_{DC}"'] & & \\
& \underset{\substack{\triangle \\ ac_{C_B}}}{C_B} \arrow[ur, "t_{CB}"] \arrow[uuu, "c_{BA}"] \arrow[rrr, phantom, "(2)", shift left=3] \arrow[rrr, phantom, "(6)"', shift right=3] & & & \underset{\substack{\triangle \\ ac_{C_D}}}{C_D} \arrow[ul, "t_{CD}"'] \arrow[lll, "c_{DB}"] \arrow[uuu, "c_{DC}"'] & \\
ac_{P_B} \triangleright P_B \arrow[ur, "b"] \arrow[uuuuu, "p_{BA}"] & & & & & P_D \lhd ac_{P_D} \arrow[lllll, "p_{DB}"] \arrow[ul, "d"'] \arrow[uuuuu, "p_{DC}"']
\end{tikzcd}
$$

- $ac_{P_D} = \bigwedge_{i \in \mathcal{I}} ac_{P_D,i}$.
  Since ac_{P_D} is a restriction of ac_{P_B} and ac_{P_C}, they must be of the form $ac_{P_B} = \bigwedge_{i \in \mathcal{I}} ac_{P_B,i}$ and $ac_{P_C} = \bigwedge_{i \in \mathcal{I}} ac_{P_C,i}$. Moreover, since ac_{P_B} and ac_{P_C} agree in ac_{P_D}, we obtain that also $ac_{P_B,i}$ and $ac_{P_C,i}$ agree in $ac_{P_D,i}$ for all $i \in \mathcal{I}$. So, by induction hypothesis, there are amalgamations $ac_{P_A,i} = ac_{P_B,i} +_{ac_{P_D,i}} ac_{P_C,i}$ such that $ac_{P_B,i}$ and $ac_{P_C,i}$ are restrictions of $ac_{P_A,i}$ for all $i \in \mathcal{I}$. Hence, $ac_{P_A} = \bigwedge_{i \in \mathcal{I}} ac_{P_A,i}$ is the required amalgamation.

- The remaining case for disjunction works analogously to the case for conjunction.

The uniqueness of the amalgamation follows from the fact that we have an amalgamation in each level of nesting and the amalgamation of typed objects is unique by Fact 4.2.

Decomposition. We do an induction over the structure of ac_{P_A}:

- $ac_{P_A} = true$.
  This case is trivial because $true = true +_{true} true$.
- $ac_{P_A} = \exists\,(a, ac_{C_A})$ with $a : P_A \to C_A$.
  Then by induction hypothesis, there exist restrictions ac_{C_B}, ac_{C_C} and ac_{C_D} of ac_{C_A} such that $ac_{C_A} = ac_{C_B} +_{ac_{C_D}} ac_{C_C}$. Moreover, by Fact 4.2, there are unique restrictions b, c and d of a such that $a = b +_d c$. Hence, we have restrictions $ac_{P_B} = \exists\,(b, ac_{C_B})$, $ac_{P_C} = \exists\,(c, ac_{C_C})$ and $ac_{P_D} = \exists\,(d, ac_{C_D})$ of ac_{P_A}, and, as shown for the case of composition before, the fact that $ac_{C_A} = ac_{C_B} +_{ac_{C_D}} ac_{C_C}$ and $a = b +_d c$ implies that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.
- $ac_{P_A} = \bigwedge_{i \in \mathcal{I}} ac_{P_A,i}$.
  Then by induction hypothesis, there exist restrictions $ac_{P_B,i}$, $ac_{P_C,i}$ and $ac_{P_D,i}$ of $ac_{P_A,i}$ such that $ac_{P_A,i} = ac_{P_B,i} +_{ac_{P_D,i}} ac_{P_C,i}$ for all $i \in \mathcal{I}$. Hence, $ac_{P_B} = \bigwedge_{i \in \mathcal{I}} ac_{P_B,i}$, $ac_{P_C} = \bigwedge_{i \in \mathcal{I}} ac_{P_C,i}$ and $ac_{P_D} = \bigwedge_{i \in \mathcal{I}} ac_{P_D,i}$ are restrictions of ac_{P_A} such that $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$.
- Again, the remaining case for disjunction works analogously to the case for conjunction.

The uniqueness of the decomposition follows from the uniqueness of restrictions by pullback construction.

□

Theorem 4.10 (Amalgamation of Solutions for Positive Nested Conditions)**.** *Given pushout (1) as in Def. 4.8 with all morphisms in $\mathcal{M}$, an amalgamation of typed objects $g_A = g_B +_{g_D} g_C$, and an amalgamation of positive nested conditions $ac_{P_A} = ac_{P_B} +_{ac_{P_D}} ac_{P_C}$ with corresponding matches $p_A = p_B +_{p_D} p_C$.*

Composition. *Given solutions Q_B for $p_B \models ac_{P_B}$ and Q_C for $p_C \models ac_{P_C}$ agreeing in a solution Q_D for $p_D \models ac_{P_D}$, then there is a solution Q_A for $p_A \models ac_{P_A}$ constructed as amalgamation $Q_A = Q_B +_{Q_D} Q_C$.*

Decomposition. *Given a solution Q_A for $p_A \models ac_{P_A}$, then there are solutions Q_B, Q_C and Q_D for $p_B \models ac_{P_B}$, $p_C \models ac_{P_C}$ and $p_D \models ac_{P_D}$, respectively, which are constructed as restrictions Q_B, Q_C and Q_D of Q_A such that $Q_A = Q_B +_{Q_D} Q_C$.*

The amalgamated composition and decomposition constructions are unique up to isomorphism.

Proof.

Composition. We perform an induction over the structure of ac_{P_A}.

- $ac_{P_A} = true$.
  Then also ac_{P_B}, ac_{P_C}, ac_{P_D} are true and we have empty solutions Q_A, Q_B, Q_C and Q_D. Since the restriction of an empty solution is empty, we have that Q_B and Q_C are restrictions of Q_A.
- $ac_{P_A} = \exists\,(a, ac_{C_A})$ with $a : P_A \to C_A$.
  By Fact 4.5 (Composition), we have the following diagram, where all rectangles are pushouts and all trapezoids are pullbacks, and all horizontal and vertical morphisms are in $\mathcal{M}$.

$$
\begin{tikzcd}
ac_{P_A} \triangleright P_A \arrow[ddr, "a"] & & & & & & P_C \triangleleft ac_{P_C} \arrow[llllll, "p_{CA}"'] \arrow[ddl, "c"'] \\
& ac_{C_A} \arrow[d, phantom, "\triangledown"] & & (2') & & ac_{C_C} \arrow[d, phantom, "\triangledown"] & \\
& C_A \arrow[dr, "t_{CA}"'] \arrow[drrr, phantom, "(2)"] & & & & C_C \arrow[llll, "c_{CA}"'] \arrow[dl, "t_{CC}"] & \\
& & TG_A & & TG_C \arrow[ll, "tg_{CA}"'] & & \\
& & TG_B \arrow[u, "tg_{BA}"] \arrow[urr, phantom, "(1)"] & & TG_D \arrow[ll, "tg_{DB}"'] \arrow[u, "tg_{DC}"'] & & \\
& C_B \arrow[ur, "t_{CB}"] \arrow[uuu, "c_{BA}"'] \arrow[uur, phantom, "(3)"] \arrow[urrr, phantom, "(4)"] & & & & C_D \arrow[llll, "c_{DB}"] \arrow[ul, "t_{CD}"'] \arrow[uuu, "c_{DC}"] \arrow[uul, phantom, "(5)"] & \\
& ac_{C_B} \arrow[u, phantom, "\triangle"] & & (4') & & ac_{C_D} \arrow[u, phantom, "\triangle"] & \\
ac_{P_B} \triangleright P_B \arrow[uuuuuuu, "p_{BA}"] \arrow[uur, "b"] \arrow[uuuuur, phantom, "(3')"] & & & & & & P_D \triangleleft ac_{P_D} \arrow[llllll, "p_{DB}"] \arrow[uul, "d"'] \arrow[uuuuuuu, "p_{DC}"'] \arrow[uuuuul, phantom, "(5')"]
\end{tikzcd}
$$

Now, we consider solutions $Q_B = (q_B, Q_{CB})$, $Q_C = (q_C, Q_{CC})$ and $Q_D = (q_D, Q_{CD})$ for $p_B \models ac_{P_B}$, $p_C \models ac_{P_C}$ and $p_D \models ac_{P_D}$, respectively, such that Q_D is a restriction of Q_B and Q_C. Then we also have that q_D is a restriction of q_B and q_C, and thus

$$g_{BA} \circ q_B \circ c_{DB} = g_{BA} \circ g_{DB} \circ q_D = g_{CA} \circ g_{DC} \circ q_D = g_{CA} \circ q_C \circ c_{DC}.$$

Together with the pushout over the Cs, this implies a unique morphism $q_A : C_A \to G_A$ with $q_A \circ c_{BA} = g_{BA} \circ q_B$ and $q_A \circ c_{CA} = g_{CA} \circ q_C$.

$$
\begin{tikzcd}
ac_{P_A} \triangleright P_A \arrow[dr, "a"] \arrow[ddrr, bend left, "p_A"] & & & & & & & & P_C \triangleleft ac_{P_C} \arrow[llllllll, "p_{CA}"'] \arrow[dl, "c"'] \arrow[ddll, bend right, "p_C"'] \\
& ac_{C_A} \triangleright C_A \arrow[dr, "q_A"] & & & & & & C_C \triangleleft ac_{C_C} \arrow[llllll, "c_{CA}"] \arrow[dl, "q_C"'] & \\
& & G_A \arrow[dr, "g_A"'] & & & & G_C \arrow[llll, "g_{CA}"] \arrow[dl, "g_C"] & & \\
& & & TG_A & & TG_C \arrow[ll, "tg_{CA}"'] & & & \\
& & & TG_B \arrow[u, "tg_{BA}"] & & TG_D \arrow[ll, "tg_{DB}"'] \arrow[u, "tg_{DC}"'] & & & \\
& & G_B \arrow[ur, "g_B"] \arrow[uuu, "g_{BA}"] & & & & G_D \arrow[llll, "g_{DB}"] \arrow[ul, "g_D"'] \arrow[uuu, "g_{DC}"'] & & \\
& ac_{C_B} \triangleright C_B \arrow[ur, "q_B"] \arrow[uuuuu, "c_{BA}"] & & & & & & C_D \triangleleft ac_{C_D} \arrow[llllll, "c_{DB}"] \arrow[ul, "q_D"'] \arrow[uuuuu, "c_{DC}"'] & \\
ac_{P_B} \triangleright P_B \arrow[uuuuuuu, "p_{BA}"] \arrow[ur, "b"] \arrow[uurr, bend right, "p_B"'] & & & & & & & & P_D \triangleleft ac_{P_D} \arrow[llllllll, "p_{DB}"] \arrow[ul, "d"'] \arrow[uull, bend left, "p_D"] \arrow[uuuuuuu, "p_{DC}"']
\end{tikzcd}
$$

Moreover, we have

$$q_A \circ a \circ p_{BA} = q_A \circ c_{BA} \circ b = g_{BA} \circ q_B \circ b = g_{BA} \circ p_B = p_A \circ p_{BA}$$

and analogously $q_A \circ a \circ p_{CA} = p_A \circ p_{CA}$. Since p_{BA} and p_{CA} are jointly epimorphic, this implies that $q_A \circ a = p_A$.

In order to show that $q_A \in \mathcal{M}$, we consider the following diagram in the left:

$$
\begin{tikzcd}
G_A & G_C \arrow[l, "g_{CA}"'] & C_C \arrow[l, "q_C"'] \\
G_B \arrow[u, "g_{BA}"] \arrow[ur, phantom, "(6)"] & G_D \arrow[l, "g_{DB}"] \arrow[u, "g_{DC}"] \arrow[ur, phantom, "(8)"] & C_D \arrow[l, "q_D"] \arrow[u, "c_{DC}"'] \\
C_B \arrow[u, "q_B"] \arrow[ur, phantom, "(7)"] & C_D \arrow[l, "c_{DB}"] \arrow[u, "q_D"] \arrow[ur, phantom, "(9)"] & C_D \arrow[l, "id_{C_D}"] \arrow[u, "id_{C_D}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
& & C_C \arrow[dll, "g_{CA} \circ q_C"'] \arrow[dl, "c_{CA}"] & \\
G_A & C_A \arrow[l, "q_A"'] & & C_D \arrow[ul, "c_{DC}"'] \arrow[dl, "c_{DB}"] \\
& & C_B \arrow[ul, "c_{BA}"'] \arrow[ull, "g_{BA} \circ q_B"] &
\end{tikzcd}
$$

We have that (6) is a pushout with all morphisms in $\mathcal{M}$ and thus also a pullback. Diagrams (7) and (8) are pullbacks by restriction, and (9) is a pullback because $q_D \in \mathcal{M}$ is a monomorphism. Hence, by composition of pullbacks, we obtain that the complete diagram is a pullback along $\mathcal{M}$-morphisms $g_{BA} \circ q_B$ and $g_{CA} \circ q_C$, which means that the pushout of the Cs is effective (see Def. 2.3), implying that $q_A \in \mathcal{M}$.
It remains to show that q_B and q_C are restrictions of q_A. In the following diagram, we have that (10) and (11) are pullbacks by restrictions, the Cs and the Gs form pushouts (see Rem. 4.9) and all morphisms in (10)-(13) are in $\mathcal{M}$. So, the horizontal as well as the vertical VK property implies that also (12) and (13) are pullbacks, which means that q_B and q_C are restrictions of q_A.

$$\begin{tikzcd}
ac_{C_A} \rhd C_A \arrow[dr, "q_A"] & & & & & C_C \lhd ac_{C_C} \arrow[lllll, "\substack{c_{CA}\\(12)}"'] \arrow[dl, "q_C"'] \\
& G_A \arrow[dr, "g_A"] & & & G_C \arrow[lll, "g_{CA}"'] \arrow[dl, "g_C"'] & \\
& & TG_A & TG_C \arrow[l, "tg_{CA}"'] & & \\
& & TG_B \arrow[u, "tg_{BA}"] & TG_D \arrow[l, "tg_{DB}"'] \arrow[u, "tg_{DC}"'] & & \\
& G_B \arrow[uuu, "g_{BA}"'] \arrow[ur, "g_B"] & & & G_D \arrow[lll, "\substack{g_{DB}\\(10)}"'] \arrow[uuu, "g_{DC}\;(11)"'] \arrow[ul, "g_D"] & \\
ac_{C_B} \rhd C_B \arrow[uuuuu, "c_{BA}\;(13)"'] \arrow[ur, "q_B"] & & & & & C_D \lhd ac_{C_D} \arrow[lllll, "c_{DB}"] \arrow[uuuuu, "c_{DC}"'] \arrow[ul, "q_D"']
\end{tikzcd}$$

Finally, Q_D being a restriction of Q_B and Q_C means that Q_{CD} is a restriction of Q_{CB} and Q_{CC} by induction hypothesis, this implies a solution Q_{CA} of $q_A \models ac_{C_A}$ such that Q_{CB} and Q_{CC} are restrictions of Q_{CA}. Hence, $Q_A = (q_A, Q_{CA})$ is a solution for $p_A \models ac_A$ such that Q_B and Q_C are restrictions of Q_A.

- $ac_{P_A} = \bigwedge_{i\in\mathcal{I}} ac_{P_A,i}$.
  We have $ac_{P_B} = \bigwedge_{i\in\mathcal{I}} ac_{P_B,i}$, $ac_{P_C} = \bigwedge_{i\in\mathcal{I}} ac_{P_C,i}$ and $ac_{P_D} = \bigwedge_{i\in\mathcal{I}} ac_{P_D,i}$ such that for all $i \in \mathcal{I}$ there is $ac_{P_D,i}$ a restriction of $ac_{P_B,i}$ and $ac_{P_C,i}$.
  Moreover, given solutions Q_B, Q_C and Q_D of $p_B \models ac_{P_B}$, $p_C \models ac_{P_C}$ and $p_D \models ac_{P_D}$, respectively, we have $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$, $Q_C = (Q_{C,i})_{i\in\mathcal{I}}$ and $Q_D = (Q_{D,i})_{i\in\mathcal{I}}$ such that for all $i \in \mathcal{I}$ we have that $Q_{B,i}$, $Q_{C,i}$ and $Q_{D,i}$ are solutions for $p_B \models ac_{P_B,i}$, $p_C \models ac_{P_C,i}$ and $p_D \models ac_{P_D,i}$, respectively, and $Q_{D,i}$ is a restriction of $Q_{B,i}$ and $Q_{C,i}$.
  Then, by induction hypothesis, there are solutions $Q_{A,i}$ for $p_A \models ac_{P_A,i}$ for all $i \in \mathcal{I}$ such that $Q_{B,i}$ and $Q_{C,i}$ are restrictions of $Q_{A,i}$. Hence, $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ is the required solution for $p_A \models ac_{P_A}$.
- $ac_{P_A} = \bigvee_{i\in\mathcal{I}} ac_{P_A,i}$.
  We have $ac_{P_B} = \bigvee_{i\in\mathcal{I}} ac_{P_B,i}$, $ac_{P_C} = \bigvee_{i\in\mathcal{I}} ac_{P_C,i}$ and $ac_{P_D} = \bigvee_{i\in\mathcal{I}} ac_{P_D,i}$ such that for all $i \in \mathcal{I}$ there is $ac_{P_D,i}$ a restriction of $ac_{P_B,i}$ and $ac_{P_C,i}$.
  Moreover, given solutions Q_B, Q_C and Q_D of $p_B \models ac_{P_B}$, $p_C \models ac_{P_C}$ and $p_D \models ac_{P_D}$, respectively. Then we have $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$, $Q_C = (Q_{C,i})_{i\in\mathcal{I}}$ and $Q_D = (Q_{D,i})_{i\in\mathcal{I}}$ such that for some $j_B, j_C, j_D \in \mathcal{I}$ we have that Q_{B,j_B}, Q_{C,j_C} and Q_{D,j_D} are solutions for $p_B \models ac_{P_B,j_B}$, $p_C \models ac_{P_C,j_C}$ and $p_D \models ac_{P_D,j_D}$, respectively, and for all $k_B, k_C, k_D \in \mathcal{I}$ with $k_B \neq j_B$, $k_C \neq j_C$ and $k_D \neq j_D$ we have that Q_{B,k_B}, Q_{C,k_C} and Q_{D,k_D} are empty. Furthermore, $Q_{D,i}$ is a restriction of $Q_{B,i}$ and $Q_{C,i}$ for all $i \in \mathcal{I}$.

  Case 1. $Q_{D,j_D} = \emptyset$.

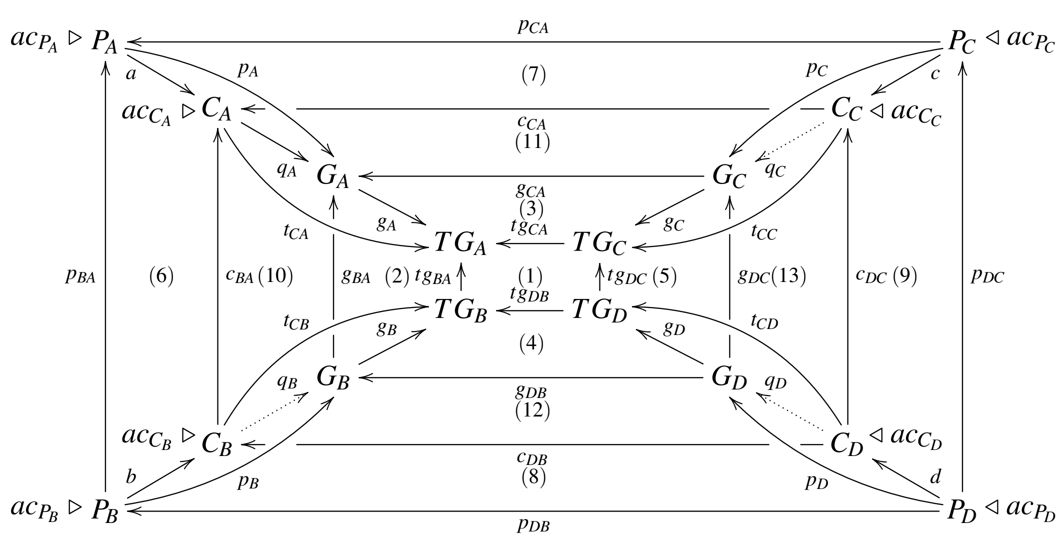
  Then we have $Q_{D,j} = \emptyset$ for all $j \in \mathcal{I}$. According to Def. 3.3, only the restriction of an empty solution is empty, implying that we also have $Q_{B,j} = Q_{C,j} = \emptyset$ for all $j \in \mathcal{I}$. Moreover, since Q_{D,j_D} is a solution for $p_D \vDash ac_{P_D,j_D}$, we can conclude that $ac_{P_D,j_D} = true$, and by the fact that ac_{P_D,j_D} is a restriction of ac_{P_A,j_D}, ac_{P_B,j_D} and ac_{P_C,j_D} it follows that also $ac_{P_A,j_D} = true$, $ac_{P_B,j_D} = true$ and $ac_{P_C,j_D} = true$. So, as shown above, there is a solution $Q_{A,j_D} = \emptyset$ for $p_A \vDash ac_{P_A,j_D}$. Hence, $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ with $Q_{A,i} = \emptyset$ for all $i \in \mathcal{I}$ is a solution for $p_A \vDash ac_{P_A}$ such that Q_B and Q_C are restrictions of Q_A.

Case 2. $Q_{D,j_D} \neq \emptyset$.
Then according to Def. 3.3, there are also $Q_{B,j_D} \neq \emptyset$ and $Q_{C,j_D} \neq \emptyset$ which means that $j_B = j_C = j_D$. So, by induction hypothesis, there is a solution Q_{A,j_D} for $p_A \vDash ac_{P_A,j_D}$ such that Q_{B,j_D} and Q_{C,j_D} are restrictions of Q_{A,j_D}. Hence, $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ with $Q_{A,k} = \emptyset$ for all $k \in \mathcal{I}$ with $k \neq j_D$ is a solution for $p_A \vDash ac_{P_A}$, and we have that Q_B and Q_C are restrictions of Q_A.

In the first case ($ac_{P_A} = true$), the uniqueness of the amalgamation follows from the fact that an empty solution can only be the restriction of another empty solution. In the second case ($ac_{P_A} = \exists\,(a, ac_{C_A})$), the uniqueness of $Q_A = (q_A, Q_{CA})$ follows from the uniqueness of q_A by universal pushout property, and by uniqueness of Q_{CA} by induction hypothesis. Finally, in the cases of conjunction and disjunction, the uniqueness of the solution follows from uniqueness of its components by induction hypothesis.

Decomposition. Again, we perform an induction over the structure of ac_{P_A}.

- $ac_{P_A} = true$.
  Then we also have that ac_{P_B}, ac_{P_C} and ac_{P_D} are true. Moreover, we have that Q_A is empty, leading to empty restrictions Q_B, Q_C and Q_D that are solutions for $p_B \vDash ac_{P_B}$, $p_C \vDash ac_{P_C}$ and $p_D \vDash ac_{P_D}$, respectively.
- $ac_{P_A} = \exists\,(a, ac_{C_A})$ with $a : P_A \to C_A$.
  Then we have $ac_{P_B} = \exists\,(b, ac_{C_B})$, $ac_{P_C} = \exists\,(c, ac_{C_C})$ and $ac_{P_D} = \exists\,(d, ac_{C_D})$. By amalgamation $g_A = g_B +_{g_D} g_C$, we have pullbacks (2)-(5) below. Moreover, by restrictions ac_{P_B}, ac_{P_C} and ac_{P_D} of ac_{P_A}, we have restrictions b, c and d of a, implying pullbacks (6)-(9) below. According to Rem. 4.6, we have an amalgamation of positive nested conditions $ac_{C_A} = ac_{C_B} +_{ac_{C_D}} ac_{C_C}$, which implies an amalgamation of typed objects $t_{CA} = t_{CB} +_{t_{CD}} t_{CC}$ by Def. 4.4.



Now, given a solution $Q_A = (q_A, Q_{CA})$ for $p_A \models ac_{P_A}$, there is $q_A : C_A \to G_A \in \mathcal{M}$ with $q_A \circ a = p_A$.
Furthermore, we have

$$g_A \circ q_A \circ c_{BA} = t_{CA} \circ c_{BA} = tg_{BA} \circ t_{CB},$$

which implies a unique morphism $q_B : C_B \to G_B$ by pullback (2) such that $g_B \circ q_B = t_{CB}$ and $g_{BA} \circ q_B = q_A \circ c_{BA}$. Due to amalgamation $t_{CA} = t_{CB} +_{t_{CD}} t_{CC}$, we have that t_{CB} is a restriction of t_{CA} and thus (10)+(2) is a pullback. So, together with pullback (2), we obtain that also (10) is a pullback by pullback decomposition and, thus, q_B is a restriction of q_A.
Moreover, by $q_A, tg_{BA} \in \mathcal{M}$ and closure of $\mathcal{M}$ under pullbacks, we know that $q_B, g_{BA} \in \mathcal{M}$. Hence, by

$$g_{BA} \circ p_B = p_A \circ p_{BA} = q_A \circ a \circ p_{BA} = q_A \circ c_{BA} \circ b = g_{BA} \circ q_B \circ b,$$

we obtain $p_B = q_B \circ b$ because $g_{BA} \in \mathcal{M}$ is a monomorphism.
Analogously, due to pullback (3) and restriction t_{CC} of t_{CA}, there is a unique restriction $q_C : C_C \to G_C \in \mathcal{M}$ of q_A with pullback (11) such that $p_C = q_C \circ c$, and due to pullback (4) and restriction t_{CD} of t_{CB}, there is a unique restriction $q_D : C_D \to G_D \in \mathcal{M}$ of q_B with pullback (12) such that $p_D = q_D \circ d$. Then, since t_{CD} is a restriction of t_{CC}, (5)+(13) is a pullback which implies that also (13) is a pullback by pullback decomposition and pullback (5). Thus, q_D is also a restriction of q_C, which means that we have $q_A = q_B +_{q_D} q_C$.
So, by induction hypothesis, there are solutions Q_{CB} for $q_B \models ac_{C_B}$, Q_{CC} for $q_C \models ac_{C_C}$, and Q_{CD} for $q_D \models ac_{C_D}$ such that $Q_{CA} = Q_{CB} +_{Q_{CD}} Q_{CC}$. Hence, for $Q_B = (q_B, Q_{CB})$, $Q_C = (q_C, Q_{CC})$ and $Q_D = (q_D, Q_{CD})$ we obtain that $Q_A = Q_B +_{Q_D} Q_C$.

- $ac_{P_A} = \bigwedge_{i\in\mathcal{I}} ac_{P_A,i}$.
  Then we also have $ac_{P_B} = \bigwedge_{i\in\mathcal{I}} ac_{P_B,i}$, $ac_{P_C} = \bigwedge_{i\in\mathcal{I}} ac_{P_C,i}$, and $ac_{P_D} = \bigwedge_{i\in\mathcal{I}} ac_{P_D,i}$. Now, given a solution $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ for $p_A \models ac_{P_A}$, then $Q_{A,i}$ is a solution for $p_A \models ac_{P_A,i}$ for all $i \in \mathcal{I}$. Thus, by induction hypothesis for all $i \in \mathcal{I}$, there are solutions $Q_{B,i}$ for $p_B \models ac_{P_B,i}$, $Q_{C,i}$ for $p_C \models ac_{P_C,i}$, and $Q_{D,i}$ for $ac_{P_D,i}$ such that $Q_{A,i} = Q_{B,i} +_{Q_{D,i}} Q_{C,i}$. This in turn means that for all $i \in \mathcal{I}$ there are $Q_{B,i}$ and $Q_{C,i}$ restrictions of $Q_{A,i}$, and $Q_{D,i}$ is a restriction of $Q_{B,i}$ and $Q_{C,i}$. Hence, for $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$, $Q_C = (Q_{C,i})_{i\in\mathcal{I}}$ and $Q_D = (Q_{D,i})_{i\in\mathcal{I}}$ we have that Q_B and Q_C are restrictions of Q_A, and Q_D is a restriction of Q_B and Q_C, implying $Q_A = Q_B +_{Q_D} Q_C$.
- $ac_{P_A} = \bigvee_{i\in\mathcal{I}} ac_{P_A,i}$.
  Then we also have $ac_{P_B} = \bigvee_{i\in\mathcal{I}} ac_{P_B,i}$, $ac_{P_C} = \bigvee_{i\in\mathcal{I}} ac_{P_C,i}$, and $ac_{P_D} = \bigvee_{i\in\mathcal{I}} ac_{P_D,i}$. Given a solution $Q_A = (Q_{A,i})_{i\in\mathcal{I}}$ for $p_A \models ac_{P_A}$, then there is $j \in \mathcal{I}$, such that $Q_{A,j}$ is a solution for $p_A \models ac_{P_A,j}$, and for all $k \in \mathcal{I}$ with $k \neq j$ there is $Q_{A,k} = \emptyset$. By induction hypothesis, there are solutions $Q_{B,j}$ for $p_B \models ac_{P_B,j}$, $Q_{C,j}$ for $p_C \models ac_{P_C,j}$, and $Q_{D,j}$ for $p_D \models ac_{P_D,j}$ such that $Q_{A,j} = Q_{B,j} +_{Q_{D,j}} Q_{C,j}$. Hence, for $Q_B = (Q_{B,i})_{i\in\mathcal{I}}$,$Q_C = (Q_{C,i})_{i\in\mathcal{I}}$ and $Q_D = (Q_{D,i})_{i\in\mathcal{I}}$, where for all $k \in \mathcal{I}$ with $k \neq j$ there is $Q_{B,k} = Q_{C,k} = Q_{D,k} = \emptyset$, we have that $Q_A = Q_B +_{Q_D} Q_C$.

The uniqueness of the solutions follows from uniqueness of restrictions by pullback constructions.

□