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
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 \( M \)-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 \( M \)-adhesive categories. An \( M \)-adhesive category consists of a category \( C \) together with a class \( M \) of monomorphisms as defined in Def. 2.1 below. The concept of \( M \)-adhesive categories generalizes that of adhesive, adhesive HLR, and weak adhesive HLR categories.

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

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

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 \( \neg ac_P \), \( \bigwedge_{i \in I} ac_{P_i} \) and \( \bigvee_{i \in I} ac_{P_i} \) are nested conditions over \( P \) for nested conditions \( ac_P \), \( ac_{P_i} \) (\( i \in I \)) over \( P \) for some index set \( I \).

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, \( \exists (a, ac) \), \( \bigwedge_{i \in I} ac_{P_i} \) and \( \bigvee_{i \in I} ac_{P_i} \), where \( I \neq \emptyset \).

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 $ac_P = \exists (a_1, \text{true}) \land \exists (a_2, \exists (a_3, \text{true}) \lor \exists (a_4, \text{true}))$.

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:

- $ac_P = \text{true}$, 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
- $ac_P = \land_{i \in \mathcal{I}} ac_{P,i}$ and for all $i \in \mathcal{I}$ holds $p \models ac_{P,i}$, or
- $ac_P = \lor_{i \in \mathcal{I}} ac_{P,i}$ and for some $i \in \mathcal{I}$ holds $p \not\models ac_{P,i}$.

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 $p : P \to G$, initial satisfaction requires just the existence of an $\mathcal{M}$-morphism $p : P \to G$ which satisfies $ac_P$.

![Figure 1: Satisfaction of nested conditions](image)

(a) Satisfaction of $ac_P$ by morphism $p$

(b) Initial satisfaction of $ac_I$

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 $G \models ac_I$ 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

\[ G \models ac_I \iff \exists p : P \to G \in \mathcal{M}. p \models ac_P \text{ (see Fig. 1(b))}. \]

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:

- $ac_P = \text{true}$ and $Q = \emptyset$, or
\[ \ac_P = \exists (a, \ac_C) \text{ with } a : P \to C \text{ and } Q = (q, Q_C) \text{ with } \mathcal{M} \text{-morphism } q : C \to G \text{ such that } q \circ a = p \text{ and } Q_C \text{ is a solution for } q \models \ac_C \text{ (see Fig. 2(a)), or} \]

\[ \ac_P = \bigwedge_{i \in I} \ac_{P,i} \text{ and } Q = (Q_i)_{i \in I} \text{ such that } Q_i \text{ is a solution for } p \models \ac_{P,i} \text{ for all } i \in I, \text{ or} \]

\[ \ac_P = \bigvee_{i \in I} \ac_{P,i} \text{ and } Q = (Q_i)_{i \in I} \text{ such that there is } j \in I \text{ with solution } Q_j \text{ for } p \models \ac_{P,j} \text{ and for all } k \in I \text{ with } k \neq j \text{ it holds that } Q_k = \emptyset. \]

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 \( b_1 \) to \( b_2 \) 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 \( b_2 \) and \( c_2 \) in \( C_3 \). Thus, \( G_A \) generally satisfies \( \ac_P \).

A corresponding solution for \( p_1 \models \ac_P \) is given by \( Q_{\text{gen}} = (Q_i)_{i \in \{1, 2\}} \) 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 \( q_1 : C_i \to G_A \) for \( i = 1, 2, 3 \) are inclusions.

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 \( i_P \models \ac_I \) is given by \( Q_{\text{init}} = (p_1, Q_{\text{gen}}) \) with \( Q_{\text{gen}} \) 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.
Fact 3.4

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

\[
\begin{array}{c}
\begin{array}{c}
T G_A \xrightarrow{t_{G_B}} T G_B \\
\downarrow t \\
G_A \xleftarrow{a} G'_A \\
\end{array}
\quad \begin{array}{c}
\begin{array}{c}
T G_A \xrightarrow{t_{G_B}} T G_B \\
\downarrow t \\
G_B \xleftarrow{b} G'_B \\
\end{array}
\end{array}
\end{array}
\]

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 $ac_{P_B}$ typed over $T G_A$ and let $T G_B$ be a restriction of it with $t : T G_B \rightarrow T G_A \in \mathcal{M}$. Then we define the restriction $ac_{P_B'} = \text{Restr}_t(ac_{P_B})$ 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., $ac_{P_B} = \exists \left( \text{Restr}_t(a), \text{Restr}_t(ac_{C_A}) \right)$, and
- the restriction of a Boolean formula is given by the restrictions of its components, i.e., $\text{Restr}_t(-ac_{P_A}) = -\text{Restr}_t(ac_{P_A})$, $\text{Restr}_t(\bigwedge_{i \in I} ac_{P_A,i}) = \bigwedge_{i \in I} \text{Restr}_t(ac_{P_A,i})$, and $\text{Restr}_t(\bigvee_{i \in I} ac_{P_A,i}) = \bigvee_{i \in I} \text{Restr}_t(ac_{P_A,i})$.

\[
\begin{array}{c}
\begin{array}{c}
T G_A \xrightarrow{t} T G_B
\end{array}
\quad \begin{array}{c}
\begin{array}{c}
C_A \xleftarrow{ac_{C_A}} P_A
\end{array}
\end{array}
\end{array}
\]

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 $T G_A$ together with a restriction $ac_{P_B}$ along $t : T G_B \rightarrow T G_A$. For a morphism $p_A : P_A \rightarrow G$ and a solution $Q_A$ for $p_A \models ac_{P_A}$, the restriction $Q_B$ of $Q_A$ along $t$, written $Q_B = \text{Restr}_t(Q_A)$, is defined inductively as follows:

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

Fact 3.4 (Restriction of Solutions for Positive Nested Conditions). Given a positive nested condition $ac_{P_A}$ and a match $p_A : P_A \rightarrow G_A$ over $T G_A$ with restrictions $ac_{P_B} = \text{Restr}_t(ac_{P_A})$, $p_B = \text{Restr}_t(p_A)$ along $t : T G_B \rightarrow T G_A$. Then for a solution $Q_A$ of $p_A \models ac_{P_A}$, there is a solution $Q_B = \text{Restr}_t(Q_A)$ 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 $T G_B \xleftarrow{t_{DB}} T G_D \xrightarrow{t_{DC}} T G_C$, with $t_{DB}, t_{DC} \in \mathcal{M}$ and typed objects $G_B \xrightarrow{g_B} T G_B$, $G_C \xrightarrow{g_C} T G_C$ and $G_D \xrightarrow{g_D} T G_D$. We say $g_B, g_C$ agree in $g_D$. If $g_D$ is a restriction of $g_B$ and $g_C$, i.e., $\text{Rest}_{g_D}(g_B) = g_D = \text{Rest}_{g_D}(g_C)$.

Given pushout (1) below with all morphisms in $\mathcal{M}$ and typed objects $g_B, g_C$ agreeing in $g_D$. A morphism $g_A : G_A \rightarrow T G_A$ 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 $g_A$.

![Diagram](image)

**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 $g_A : G_A \rightarrow T G_A$, 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 $g_A : G_A \rightarrow T G_A$, 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 $g_A : G_A \rightarrow T G_A$ and $T G_A, T G_B, T G_C, T G_D$ are given such that (1) is a pushout with $M$-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 pullbacks up to isomorphism. 

Example 4.3 (Amalgamation of Typed Objects). Figure 3 shows a pushout of type graphs $T G_A$, $T G_B$, $T G_C$ and $T G_D$.

**Composition.** Consider the typed graphs $G_B$, $G_C$ and $G_D$ typed over $T G_B$, $T G_C$ and $T G_D$, 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 $T G_A$. We can restrict $G_A$ to the type graphs $T G_B$ and $T G_C$, 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 $T G_D$, 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](image)

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 $ac_{PB}$ typed over $T G_B$ and $ac_{PC}$ typed over $T G_C$ agree in $ac_{PD}$ typed over $T G_D$ if $ac_{PD}$ is a restriction of $ac_{PB}$ and $ac_{PC}$.

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

![Diagram](image)
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 \( ac_{P_A} \) and \( ac_{P_C} \) typed over \( TG_B \) and \( TG_C \), respectively, agreeing in \( ac_{P_B} \) typed over \( TG_D \), then there exists a unique positive nested condition \( ac_{P_A} \) typed over \( TG_A \) such that \( ac_{P_A} = ac_{P_B} + ac_{P_D} \). 

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

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

**Remark 4.6.** Given an amalgamation \( ac_{P_A} = ac_{P_B} + ac_{P_D} \) 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.

**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 \( ac_{P_A} = \bigvee_{i\in\{1,2\}} ac_{C_i} \) with \( ac_{C_i} = \exists (a_i, true) \) for \( i = 1, 2 \), and \( ac_{P_B} \), \( ac_{P_D} \) and \( ac_{P_P} \) have a similar structure.

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

![Figure 4: Amalgamation of positive nested conditions](image-url)
This means that we have an amalgamation $t_{P_A} = t_{P_B} + t_{P_C}$ with pushout of the $P$s as shown in Fig. 4 as well as amalgamations of the corresponding type morphisms of the $G$s, leading to the pushouts depicted in Fig. 4 by dotted arrows for the $C_1$s and by dashed arrows for the $C_2$s. The morphisms $a_{1A}$ and $a_{2A}$ are obtained by the universal property of pushouts.

**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 $p_B \vdash ac_{P_B}$ and $Q_C$ for $p_C \vdash ac_{P_C}$ agree in a solution $Q_D$ for $p_D \vdash ac_{P_D}$ if $Q_D$ is a restriction of $Q_B$ and $Q_C$.

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

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

**Decomposition.** Given a solution $Q_A$ for $p_A \vdash ac_{P_A}$, then there are solutions $Q_B$, $Q_C$ and $Q_D$ of $Q_A$ such that $Q_A = Q_B + g_D Q_C$. The amalgamated composition and decomposition constructions are unique up to isomorphism.

**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 $G$s, and by $ac_{P_A} = ac_{P_B} + ac_{P_D} ac_{P_C}$ we also have a pushout over the $P$s.

**Remark 4.11.** From the proof of Thm. 4.10 (see App. A) we can conclude that for a given amalgamation of solutions $Q_A = Q_B + g_D Q_C$, 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 $g_A = g_B +_{g_B} g_C$, and an amalgamation of positive constraints $ac_A = ac_B + ac_D ac_C$. Then we have:

**Decomposition.** Given a solution $Q_A$ for $G_A \models ac_A$, then there are solutions $Q_B$ for $G_B \models ac_B$, $Q_C$ for $G_C \models ac_C$ and $Q_D$ for $G_D \models ac_D$ such that $Q_A = Q_B + _{G_B} Q_C$.

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

**Proof.**

**Decomposition.** By Def. 2.8 a solution $Q_A$ for $G_A \models ac_A$ 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 $g_A = g_B +_{g_B} g_C$ 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 $Q_A = Q_B + _{G_B} Q_C$, and by Def. 2.8 $Q_B$, $Q_C$ and $Q_D$ are solutions for $G_B \models ac_B$, $G_C \models ac_C$ and $G_D \models ac_D$, respectively.

**Composition.** Now, given solutions $Q_B$, $Q_C$ and $Q_D$ for $G_B \models ac_B$, $G_C \models ac_C$ and $G_D \models ac_D$, 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 $i_{G_A} \vdash ac_A$ such that $Q_A = Q_B +_{Q_D} Q_C$. Finally, Def. 2.8 implies that $Q_A$ is a solution for $G_A \vdash ac_A.

\[\square\]

**Corollary 5.2** (Compatibility of Initial Satisfaction with Restriction). Given type restriction $t : TG_B \rightarrow TG_A \in \mathcal{M}$, 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 $G_A \vdash ac_A$ implies $G_B \vdash ac_B$. Moreover, if $Q_A$ is a solution for $G_A \vdash ac_A$ then $Q_B = \text{Restr}_I(Q_A)$ is a solution for $G_B \vdash ac_B$.

**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 $g_A = g_B +_{g_D} g_C$ and, analogously, $ac_A = ac_B +_{ac_D} ac_C$ with corresponding matches $i_{G_A} = i_{G_B} +_{i_{G_D}} i_{G_C}$. So, given a solution $Q_A$ for $G_A \vdash ac_A$, by item 1 of Thm. 5.1 there is a solution $Q_B$ for $G_B \vdash ac_B$ with $Q_A = Q_B +_{Q_D} Q_C$ such that by Def. 4.4 $Q_B$ is a restriction of $Q_A$. \[\square\]

**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 $ac_A = ac_B +_{ac_C} ac_C$. Note that we have $ac_A = \exists (i_{p_A}, ac_P)$ and $ac_B, ac_C$ and $ac_D$ with similar structure, where the amalgamation $ac_P = ac_B +_{ac_D} ac_C$ is presented in Ex. 4.7.

**Composition.** For $G_B \vdash ac_B$ 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 $G_C \vdash ac_C$ and $Q_D$ for $G_D \vdash ac_D$. 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 $G_A \vdash ac_A$.

**Decomposition.** For $G_A \vdash ac_A$ 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 $G_B \vdash ac_B$, $Q_C$ for $G_C \vdash ac_C$, and $Q_D$ for $G_D \vdash ac_D$ 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.
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 $ac_{P_A}$ of constraint $ac_{P_B}$. There are two possible matches $p_{1A}, p_{2A} : P_A \rightarrow G_A \in \mathcal{M}$ where $p_{1A}$ is an inclusion and $p_{2A}$ maps $v_1$ to $v_2$ and $c_1$ to $c_2$. 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 $p_{1A}$ we have $q_{1A}$ with $q_{iA} \circ a_A = p_{iA}$ for $i = 1, 2$. Thus, we have that $G_A \models ac_{P_A}$.

For the constraint $ac_{P_B}$ there is a match $p_B : P_B \rightarrow G_B \in \mathcal{M}$ mapping edge $v_1$ 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 $p_A : P_A \rightarrow G_A \in \mathcal{M}$ 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. focusing 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] 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


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 $p_A : P_A \rightarrow G_A$ over $T G_A$ with restrictions $ac_{P_B} = \text{Restr}_t(ac_{P_A})$, $p_B = \text{Restr}_t(p_A)$ along $t : T G_B \rightarrow T G_A$. Then for a solution $Q_A$ of $p_A \models ac_{P_A}$ there is a solution $Q_B = \text{Restr}_t(Q_A)$ for $p_B \models ac_{P_B}$.

Proof.

• For $ac_{P_A} = \text{true}$ 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 $ac_{P_A} = \exists (a, acc_a)$ we have that $Q_A = (q_A, Q_{CA})$ such that $q_A : C_A \rightarrow G_A \in \mathcal{M}$ with $q_A \circ a = p_A$ and $Q_{CA}$ is a solution for $q_A \models acc_a$. Thus by $q_B = \text{Restr}_t(q_A) : C_B \rightarrow G_B$, we have $q_B \in \mathcal{M}$ and we also have $t_G : G_B \rightarrow G_A \in \mathcal{M}$, because $t \in \mathcal{M}$ (see Fig. 7). So for $ac_{P_B} = \exists (b, acc_b)$ 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 $q_A \models acc_a$ implies that $Q_{CB} = \text{Restr}_t(Q_{CA})$ is a solution for $q_B \models acc_b$ 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 $ac_{P_B} = \bigwedge_{i \in \mathcal{I}} \text{Restr}_t(ac_{P_{A_i}})$. 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 $Q_{B_{A_i}} = \text{Restr}_t(Q_{A_i})$ that are solutions for $p_B \models \text{Restr}_t(ac_{P_{A_i}})$ for all $i \in \mathcal{I}$. Hence, the restriction $Q_B = (Q_{B_{A_i}})_{i \in \mathcal{I}}$ 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 $ac_{P_B} = \bigvee_{i \in \mathcal{I}} \text{Restr}_t(ac_{P_{A_i}})$. 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 $p_B \models \text{Restr}_t(ac_{P_{A,j}})$. Hence, we also have that the restriction $Q_B = (Q_{B_{A_i}})_{i \in \mathcal{I}}$ is a solution for $p_B \models ac_{P_B}$ with $Q_{B,k} = \emptyset$ for $k \neq j$.

\[ \Box \]

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

![Figure 7: Restriction of solution $q_A$ for $p_A \models \exists (a, acc_a)$](image-url)
**Composition.** If there are positive nested conditions $ac_{P_B}$ and $ac_{P_C}$ typed over $T G_B$ and $T G_C$, respectively, agreeing in $ac_{P_b}$ typed over $T G_D$ then there exists a unique positive nested condition $ac_{P_A}$ typed over $T G_A$ 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 $T G_A$, 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 $ac_{P_B}$:

- $ac_{P_D} = true$.
  Then we also have $ac_{P_B} = true$ and $ac_{P_C} = true$, and the amalgamation $ac_{P_A}$ is trivially given by $ac_{P_A} = true$.

- $ac_{P_B} = \exists (d, ac_{C_D})$ with $d : P_D \to C_D$.
  The assumption that $ac_{P_B}$ and $ac_{P_C}$ agree in $ac_{P_B}$ means that $ac_{P_B}$ 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 $ac_{C_B}$ 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 $M$ under pullbacks, we obtain from $t_{GBA}, t_{GCA} \in M$ that also $c_{BA}, c_{CA} \in M$.

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 $C$s, leading to an amalgamation $a = b + c : P_A \to C_A$ with pullbacks (8) and (9) by Fact 4.2. Hence, $ac_{P_A} = \exists (a, ac_{C_A})$ is the required amalgamation.

- $ac_{P_D} = \wedge_{i \in I} ac_{P_B,i}$.
  Since $ac_{P_B,i}$ is a restriction of $ac_{P_B}$ and $ac_{P_C}$, they must be of the form $ac_{P_B,i} = \wedge_{i \in I} ac_{P_B,i}$ and $ac_{P_C,i} = \wedge_{i \in I} ac_{P_C,i}$.
  Moreover, since $ac_{P_B,i}$ and $ac_{P_C,i}$ agree in $ac_{P_B,i}$, we obtain that also $ac_{P_B,i}$ and $ac_{P_C,i}$ agree in $ac_{P_B,i}$ for all $i \in I$. 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 $ac_{P_A,i}$ and $ac_{P_C,i}$ are restrictions of $ac_{P_A,i}$ for all $i \in I$.
  Hence, $ac_{P_A} = \wedge_{i \in I} ac_{P_A,i}$ 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 \textbf{Fact 4.2}.

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

- $ac_{P_A} = true$.
  
  This case is trivial because $true = true + true$.
- $ac_{P_A} = \exists (a, ac_{C_A})$ with $a : P_A \rightarrow C_A$.
  
  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 \textbf{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 $ac_{P_A i}, ac_{P_B i}$ and $ac_{P_A i}$ of $ac_{P_A i}$ such that $ac_{P_A i} = ac_{P_B i} + ac_{P_D i} ac_{P_C}$ 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.

\begin{theorem}[Amalgamation of Solutions for Positive Nested Conditions] \label{thm:amalgamation}
Given pushout (1) as in \textbf{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 $p_A = p_B + p_D p_C$.

\textbf{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 $Q_A = Q_B + Q_D Q_C$.

\textbf{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.

\end{theorem}

\textbf{Proof.}

\textbf{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 $a : P_A \rightarrow C_A$.
  
  By \textbf{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 $\mathcal{M}$.
In order to show that $q_A$ implies that $q_C$, together with the pushout over the $C$s and analogously $q_B$, $p_c \vdash ac_P$, and $p_D \vdash ac_P$, 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 $q_A : C_A \rightarrow G_A$ with $q_A \circ c_{BA} = g_{BA} \circ q_B$ and $q_A \circ c_{CA} = g_{CA} \circ q_C$.

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:
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 $C$s 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 $C$s and the $G$s 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 $q_A$.

Finally, $Q_B$ 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 $q_A \sqsupset acc_A$ such that $Q_CB$ and $Q_CC$ are restrictions of $Q_CA$. Hence, $Q_A = (q_A, Q_CA)$ is a solution for $p_A \sqsupset ac_A$ such that $Q_B$ and $Q_C$ are restrictions of $Q_A$.

- $ac_{P_B} = \bigcap_{i \in I} ac_{P_B,i}$
  We have $ac_{P_B} = \bigcap_{i \in I} ac_{P_B,i}$, $ac_{P_C} = \bigcap_{i \in I} ac_{P_C,i}$ and $ac_{P_D} = \bigcap_{i \in I} ac_{P_D,i}$ such that for all $i \in I$ there is $ac_{P_B,i}$ a restriction of $ac_{P_B,i}$ and $ac_{P_C,i}$.

Moreover, given solutions $Q_B$, $Q_C$ and $Q_D$ of $p_B \sqsubset ac_{P_B}$, $p_C \sqsubset ac_{P_C}$ and $p_D \sqsubset ac_{P_D}$, respectively, we have $Q_B = (Q_B,i)_{i \in I}$, $Q_C = (Q_C,i)_{i \in I}$ and $Q_D = (Q_D,i)_{i \in I}$ such that for all $i \in I$ we have that $Q_{B,i}$, $Q_{C,i}$ and $Q_{D,i}$ are solutions for $p_B \sqsubset ac_{P_B,i}$, $p_C \sqsubset ac_{P_C,i}$ and $p_D \sqsubset ac_{P_D,i}$, 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 $p_A \sqsubset ac_{P_A,i}$ for all $i \in I$ such that $Q_{B,i}$ and $Q_{C,i}$ are restrictions of $Q_{A,i}$. Hence, $Q_A = (Q_{A,i})_{i \in I}$ is the required solution for $p_A \sqsubset ac_{P_A}$.

- $ac_{P_A} = \bigvee_{i \in I} ac_{P_A,i}$
  We have $ac_{P_B} = \bigvee_{i \in I} ac_{P_B,i}$, $ac_{P_C} = \bigvee_{i \in I} ac_{P_C,i}$ and $ac_{P_D} = \bigvee_{i \in I} ac_{P_D,i}$ such that for all $i \in I$ there is $ac_{P_B,i}$ a restriction of $ac_{P_B,i}$ and $ac_{P_C,i}$.

Moreover, given solutions $Q_B$, $Q_C$ and $Q_D$ of $p_B \sqsubset ac_{P_B}$, $p_C \sqsubset ac_{P_C}$ and $p_D \sqsubset ac_{P_D}$, respectively. Then we have $Q_B = (Q_B,i)_{i \in I}$, $Q_C = (Q_C,i)_{i \in I}$ and $Q_D = (Q_D,i)_{i \in I}$ such that for some $j_B,j_C,j_D \in I$ we have that $Q_{B,j_B}$, $Q_{C,j_C}$ and $Q_{D,j_D}$ are solutions for $p_B \sqsubset ac_{P_B,j_B}$, $p_C \sqsubset ac_{P_C,j_C}$ and $p_D \sqsubset ac_{P_D,j_D}$, respectively, and for all $k_B,k_C,k_D \in I$ 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 $i \in I$.

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

Then we have $Q_{D,j} = \emptyset$ for all $j \in I$. 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 \( Q_{D,j} \) is a solution for \( p_D \models ac_{P_D,j} \), we can conclude that \( ac_{P_D,j} = true \), and by the fact that \( ac_{P_A,j} \) is a restriction of \( ac_{P_A,j} \), \( ac_{P_B,j} \) and \( ac_{P_C,j} \) it follows that also \( ac_{P_A,j} = true \), \( ac_{P_B,j} = true \) and \( ac_{P_C,j} = true \). So, as shown above, there is a solution \( Q_{A,j} = \emptyset \) for \( p_A \models ac_{P_A,j} \). 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 \( p_A \models ac_{P_A} \) such that \( Q_B \) and \( Q_C \) are restrictions of \( Q_A \).

**Case 2.** \( Q_{D,j} \neq \emptyset \).

Then according to Def. 3.3 there are also \( Q_{B,j} \neq \emptyset \) and \( Q_{C,j} \neq \emptyset \) which means that \( j_B = j_C = j_D \). So, by induction hypothesis, there is a solution \( Q_{A,j} \) for \( p_A \models ac_{P_A,j} \) such that \( Q_{B,j} \) and \( Q_{C,j} \) are restrictions of \( Q_{A,j} \). 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 \( p_A \models ac_{P_A} \), 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 \( p_B \models ac_{P_B}, p_C \models ac_{P_C} \) and \( p_D \models ac_{P_D} \), respectively.

- \( ac_{P_A} = \exists (a,ac_{C_A}) \) with \( a : P_A \rightarrow C_A \).
  
  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 \( q_A = q_B + q_D q_C \), 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 $p_A \vdash acp_A$, there is $q_A : C_A \to G_A \in \mathcal{M}$ with $q_A \circ a = p_A$.

Furthermore, we have

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

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 $t_{CA} = t_{CB} + t_{CD}$, 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 $q_A, t_{GB} \in \mathcal{M}$ and closure of $\mathcal{M}$ under pullbacks, we know that $q_B, g_{BA} \in \mathcal{M}$. Hence, by

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

we obtain $p_B = q_B \circ b$ because $g_{BA} \in \mathcal{M}$ is a monomorphism.

Analogously, due to pullback (3) and restriction $t_{CD}$ 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 $p_C = q_C \circ c_C$, 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 $q_B \vdash ac_{CB}$, $Q_{CC}$ for $q_C \vdash ac_{CC}$, and $Q_{CD}$ for $q_D \vdash ac_{CD}$ 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 $Q_A = Q_B + Q_D Q_C$.

• $acp_A = \bigwedge_{i \in I} acp_{A,i}$.

Then we also have $acp_B = \bigwedge_{i \in I} acp_{B,i}$, $acp_C = \bigwedge_{i \in I} acp_{C,i}$, and $acp_D = \bigwedge_{i \in I} acp_{D,i}$. Now, given a solution $Q_A = (Q_{A,i})_{i \in I}$ for $p_A \vdash acp_A$, then $Q_{A,i}$ is a solution for $p_A \vdash acp_{A,i}$ for all $i \in I$. Thus, by induction hypothesis for all $i \in I$, there are solutions $Q_{B,i}$ for $p_B \vdash acp_{B,i}$, $Q_{C,i}$ for $p_C \vdash acp_{C,i}$, and $Q_{D,i}$ for $acp_{D,i}$ such that $Q_{A,i} = Q_{B,i} + Q_{D,i} Q_{C,i}$. This in turn means that for all $i \in I$ 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 $Q_B = (Q_{B,i})_{i \in I}$, $Q_C = (Q_{C,i})_{i \in I}$ and $Q_D = (Q_{D,i})_{i \in I}$ 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 $Q_A = Q_B + Q_D Q_C$.

• $acp_A = \bigvee_{i \in I} acp_{A,i}$.

Then we also have $acp_B = \bigvee_{i \in I} acp_{B,i}$, $acp_C = \bigvee_{i \in I} acp_{C,i}$, and $acp_D = \bigvee_{i \in I} acp_{D,i}$. Given a solution $Q_A = (Q_{A,i})_{i \in I}$ for $p_A \vdash acp_A$, then there is $j \in I$, such that $Q_{A,j}$ is a solution for $p_A \vdash acp_{A,j}$, and for all $k \in I$ with $k \neq j$ there is $Q_{A,k} = \emptyset$. By induction hypothesis, there are solutions $Q_{B,j}$ for $p_B \vdash acp_{B,j}$, $Q_{C,j}$ for $p_C \vdash acp_{C,j}$, and $Q_{D,j}$ for $p_D \vdash acp_{D,j}$ such that $Q_{A,j} = Q_{B,j} + Q_{D,j} Q_{C,j}$. Hence, for $Q_B = (Q_{B,i})_{i \in I}$, $Q_C = (Q_{C,i})_{i \in I}$ and $Q_D = (Q_{D,i})_{i \in I}$, where for all $k \in I$ 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.

□