Satisfiability of High-Level Conditions

Annegret Habel and Karl-Heinz Pennemann

Carl v. Ossietzky Universität Oldenburg, Germany**
{habel,pennemann}@informatik.uni-oldenburg.de

Abstract. In this paper, we consider high-level structures like graphs, Petri nets, and algebraic specifications and investigate two kinds of satisfiability of conditions and two kinds of rule matching over these structures. We show that, for weak adhesive HLR categories with class $\mathcal{A}$ of all morphisms and a class $\mathcal{M}$ of monomorphisms, strictly closed under decompositions, $\mathcal{A}$- and $\mathcal{M}$-satisfiability and $\mathcal{A}$- and $\mathcal{M}$-matching are expressively equivalent. The results are applied to the category of graphs, where $\mathcal{M}$ is the class of all injective graph morphisms.

1 Introduction

Conditions are most important for high-level systems in a large variety of application areas, especially in the area of safety-critical systems e.g. the specification of railroad control systems [8] and access control policies [11]. Conditions are properties on morphisms or objects which have to be satisfied.

Adhesive HLR systems, introduced in [6] are a new version of high-level replacement systems, combining HLR systems in the sense of [5] and adhesive categories [12]. Weak adhesive HLR categories consist of a category and a class $\mathcal{M}$ of monomorphisms and can be applied to all kinds of graphs, Petri nets, and algebraic specifications.

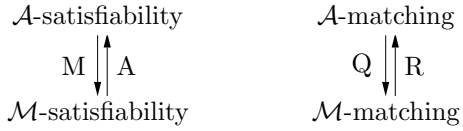
In this paper, we investigate rules and conditions on high-level structures and speak on $\mathcal{A}$-matching if the match of the rule is arbitrary and on $\mathcal{M}$-matching if the match is in $\mathcal{M}$. Accordingly, we speak on $\mathcal{A}$-satisfiability of a condition if the required morphisms are arbitrary and on $\mathcal{M}$-satisfiability if they are in $\mathcal{M}$. In the literature, nearly every combination of matching and satisfiability occurs. The different concepts have their advantages and disadvantages: $\mathcal{A}$-matching and $\mathcal{A}$-satisfiability allow a compact representation of a set of similar rules and conditions, but this might be a source of error. $\mathcal{M}$-matching and $\mathcal{M}$-satisfiability restrict the allowed morphisms, allow counting, are flexible and intuitive, but one may need sets of similar rules and conditions.

We systematically investigate the matching and satisfiability notions with the aim to find a simple transformation for switching from one notion to the other.

** This work is supported by the German Research Foundation (DFG), grants GRK 1076/1 (Graduate School on Trustworthy Software Systems) and HA 2936/2 (Development of Correct Graph Transformation Systems).

We show how to transform conditions from $\mathcal{A}$- to $\mathcal{M}$-satisfiability such that a morphism $\mathcal{A}$-satisfies a condition if and only if it $\mathcal{M}$-satisfies the transformed condition and from $\mathcal{M}$- to $\mathcal{A}$-satisfiability such that a morphism $\mathcal{M}$-satisfies a condition if and only if it $\mathcal{A}$-satisfies the transformed condition, provided that the class $\mathcal{M}$ is strictly closed under decompositions. This is an important step for connecting conditions traditionally considered with $\mathcal{M}$-satisfiability with Rensink's graph predicates [13] and first order formulas, whose satisfiability notions have to be classified as $\mathcal{A}$-satisfiability: E.g., the semantics of formulas is concerned with arbitrary assignments of variables to values, i.e. an assignment is a not necessarily injective function from variables to the domain.

Moreover, we investigate $\mathcal{M}$-matching of rules in the framework of weak adhesive HLR categories and present two Simulation Theorems, saying that direct derivations can be simulated by direct derivations with $\mathcal{M}$-matching and vice versa. The transformations are illustrated by examples in the category of graphs where $\mathcal{M}$ is the class of all injective graph morphisms.



The paper is organized as follows. In Section 2, we recall the notions of conditions, rules, and direct derivations in the framework of weak adhesive HLR categories. In Section 3, we show that, for weak adhesive HLR categories with $\mathcal{M}$-initial object and $\mathcal{M}$ strictly closed under decompositions, $\mathcal{A}$-satisfiability and $\mathcal{M}$-satisfiability are expressively equivalent. In Section 4, we distinguish between $\mathcal{A}$-matching, where the match is allowed to be arbitrary, and $\mathcal{M}$-matching, where the match must be in $\mathcal{M}$, and show that, for weak adhesive HLR categories with $\mathcal{M}$ strictly closed under decompositions, $\mathcal{A}$-matching and $\mathcal{M}$-matching are expressively equivalent. A conclusion including further work is given in Section 5.

2 Conditions and rules

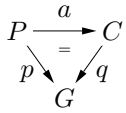
In this section, we review the definitions of conditions and rules for high-level structures like graphs, Petri nets, and algebraic specifications. We use the framework of weak adhesive HLR categories. For a detailed introduction see [6, 4].

Assumption. We assume that $\langle \mathcal{C}, \mathcal{M} \rangle$ is a weak adhesive HLR category.

Example 1. The category $\langle \mathbf{Graphs}, \mathcal{M} \rangle$ of directed graphs, where $\mathcal{M}$ is the class of all injective graph morphisms, is a weak adhesive HLR category.

Definition 1 (conditions). A *condition* over an object P is of the form $\exists a$ or $\exists(a, c)$ where $a\colon P \to C$ is a morphism and c is a condition over C. Moreover, Boolean formulas over conditions [over P] are conditions [over P]. A morphism $p\colon P \to G$ *satisfies* a condition $\exists a$ $[\exists(a, c)]$ if there exists a morphism $q\colon C \to G$

in $\mathcal{M}$ with $q \circ a = p$ [satisfying c]. An object G *satisfies* a condition $\exists a$ [$\exists(a, c)$] if all morphisms $p\colon P \to G$ in $\mathcal{M}$ satisfy the condition.



The satisfaction of conditions is extended onto Boolean conditions in the usual way. We write $p \models_{\mathcal{M}} c$ or $p \models c$ [$G \models c$] to denote that morphism p [object G] satisfies c. Two conditions c and c' over P are *equivalent* on morphisms, denoted by $c \equiv c'$, if, for all morphisms $p\colon P \to G$, $p \models c$ if and only if $p \models c'$.

Remark 1. The conditions in Definition 1 correspond to nested constraints and application conditions in [8] and they subsume the previous notions of constraints and application conditions in [3].

In the definition, the required morphisms have to be in $\mathcal{M}$. We sometimes speak of $\mathcal{M}$-satisfiability. Besides $\mathcal{M}$-satisfiability, one may investigate $\mathcal{A}$-satisfiability, where the required morphisms are allowed to be arbitrary, i.e. in $\mathcal{A}$, where $\mathcal{A}$ denotes the class of all morphisms. The definition is obtained from the one of $\mathcal{M}$-satisfiability, by replacing all occurrences of $\mathcal{M}$ by $\mathcal{A}$ resp. deleting all occurrences of "in $\mathcal{M}$". We write $\models_{\mathcal{A}}$ to denote $\mathcal{A}$-satisfiability.

Conditions of the form $\exists\,\mathrm{id}$ and $\neg\exists\,\mathrm{id}$ with identity $\mathrm{id}\colon P \to P$ are abbreviated by true and false, respectively. It turns out that, for every condition c, $\exists(\mathrm{id}, c) \equiv c$.

Example 2. In the category of graphs, the meaning of the following conditions for a graph morphism w.r.t. $\mathcal{M}$-satisfiability is:

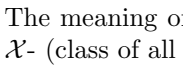
$\neg\exists(\underset{1}{\circ}\ \underset{2}{\circ} \to \underset{1}{\circ}\rightrightarrows\underset{2}{\circ})$	There do not exist parallel edges between the images of $1, 2$.
$\exists(\underset{1}{\circ} \to S_n)$	The image of 1 has n outgoing edges to different nodes. S_n denotes a star with n outgoing edges.
$\neg\exists(\underset{1}{\circ} \to S_{n+1})$	The image of 1 does not have $n{+}1$ outg. edges to diff. nodes.
$\exists(\underset{1}{\circ}\ \underset{2}{\circ} \to P_n)$	There is a simple path of length n connecting the im. of $1, 2$. P_n denotes a simple path of length n between the im. of $1, 2$.

$\mathcal{M}$-satisfiability restricts the kind of morphisms: no identification of nodes and edges is allowed. If one wants to have a condition c with arbitrary satisfiability, one has to add the so-called quotient conditions of c to the system. $\mathcal{M}$-satisfiability allows to express explicit counting such as the existence/non-existence of n nodes or edges or a simple path of length n.

The meaning of the condition for a graph morphism w.r.t. $\mathcal{A}$-satisfiability is:

$\exists(\underset{1}{\circ} \to \underset{1}{\circ}\to\circ)$	There exists an outgoing edge (proper edge or loop).

$\mathcal{A}$-satisfiability allows a compact representation of a set of similar conditions. On the other hand, $\mathcal{A}$- satisfiability may lead to misinterpretations of conditions.

The meaning of the condition $\exists($  $)$ for a graph morphism w.r.t. $\mathcal{M}$-, $\mathcal{X}$- (class of all edge-injective graph morphisms), and $\mathcal{A}$-satisfiability is:

> There exist at least two proper outgoing edges to different nodes.
> There exist at least two outgoing edges (proper edges and/or loops).
> There exists at least one outgoing edge (proper edge or loop).

$\mathcal{A}$-satisfiability may be seen as the complement to $\mathcal{M}$-satisfiability. Both notions have their advantages as well as their disadvantages: E.g. $\mathcal{M}$-satisfiability allows explicit counting, but conditions get complex, if it is undesired to distinguish elements. For each notion there exist examples of properties, for which the conditions get complex when expressed by the other notion, so none is better than the other.

We consider rules with application conditions [3, 8]. Examples and pointers to the literature can be found in [2, 1, 7].

Definition 2 (rules). A *plain rule* $q = \langle L \leftarrow K \rightarrow R \rangle$ consists of two morphisms in $\mathcal{M}$ with a common domain K. L is called the left-hand side, R the right-hand side, and K the interface. An *application condition* $\mathrm{ac} = \langle \mathrm{ac}_L, \mathrm{ac}_R \rangle$ for q consists of two conditions over L and R, respectively. A *rule* $p = \langle q, \mathrm{ac} \rangle$ consists of a plain rule q and an application condition ac for q.

$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ {\scriptstyle m}\downarrow & (1) & \downarrow & (2) & \downarrow{\scriptstyle m^*} \\ G & \longleftarrow & D & \longrightarrow & H \end{array}$$

Given a plain rule q and a morphism $K \rightarrow D$, a *direct derivation* consists of two pushouts (1) and (2). We write $G \Rightarrow_{q,m,m^*} H$, $G \Rightarrow_q H$, or short $G \Rightarrow H$ and say that m is the *match* and m^* is the *comatch* of q in H. We speak of an $\mathcal{A}$-match ($\mathcal{A}$-matching) if m is arbitrary and of an $\mathcal{M}$-match ($\mathcal{M}$-matching) if m is in $\mathcal{M}$. Given a rule $p = \langle q, \mathrm{ac} \rangle$, there is a *direct derivation* $G \Rightarrow_{p,m,m^*} H$ in $\langle X, Y \rangle$ with X, Y in $\{\mathcal{A}, \mathcal{M}\}$ if $G \Rightarrow_{q,m,m^*} H$, $m \in X$, $m \models_Y \mathrm{ac}_L$, and $m^* \models_Y \mathrm{ac}_R$. Given a set of rules $\mathcal{R}$, we write $G \Rightarrow_{\mathcal{R}} H$ if there is a rule p in $\mathcal{R}$ such that $G \Rightarrow_p H$.

Example 3. The meaning of the following rules w.r.t. $\mathcal{M}$-matching is:

Add	$= \langle \bigcirc\ \bigcirc \leftarrow \bigcirc\ \bigcirc \rightarrow \bigcirc\!\!\rightarrow\!\!\bigcirc \rangle$	Addition of a proper edge.
Delete	$= \langle \bigcirc\!\!\rightarrow\!\!\bigcirc \leftarrow \bigcirc\ \bigcirc \rightarrow \bigcirc\ \bigcirc \rangle$	Deletion of a proper edge.
Delete_2	$= \langle \bigcirc\rightleftarrows\bigcirc \leftarrow \bigcirc\ \bigcirc \rightarrow \bigcirc\ \bigcirc \rangle$	Deletion of two parallel proper edges.

where an edge is *proper* if its source and target are different.

$\mathcal{M}$-matching restricts the applicability of the rule: no identification of nodes and edges is allowed. The addition and deletion of a loop, for example, requires to explicitly consider the corresponding quotient rules of Add and Delete. Moreover, $\mathcal{M}$-matching allows explicit counting such as the deletion of n nodes or edges.

The meaning of the following rules w.r.t. $\mathcal{A}$-matching is:

Add	$= \langle \bigcirc\ \bigcirc \leftarrow \bigcirc\ \bigcirc \rightarrow \bigcirc\!\!\rightarrow\!\!\bigcirc \rangle$	Addition of an edge (proper edge or loop).
Delete	$= \langle \bigcirc\!\!\rightarrow\!\!\bigcirc \leftarrow \bigcirc\ \bigcirc \rightarrow \bigcirc\ \bigcirc \rangle$	Deletion of an edge (proper edge or loop).
Delete_2	$= \langle \bigcirc\rightrightarrows\bigcirc \leftarrow \bigcirc\ \bigcirc \rightarrow \bigcirc\ \bigcirc \rangle$	Deletion of two edges (proper or loops).

$\mathcal{A}$-matching allows a compact representation of a set of similar rules. Every rule represents a finite set of quotient rules; the quotient rules are implicitly in the system. This advantage may become a disadvantage whenever one forgets that the rules may be applied non-injective. Additionally, identifications of elements may only take place, if all elements involved are preserved. This corresponds to an additional implicit application condition and which is often forgotten.

In general, there are several cases where $\mathcal{M}$-matching is useful, e.g. no identification, several cases where $\mathcal{A}$-matching is desired, e.g. arbitrary identification of preserved elements, several important "mixed" cases, e.g. certain elements are possibly identified, others are not, and cases which are covered by neither $\mathcal{M}$- nor $\mathcal{A}$-matching, e.g. arbitrary identification of all elements.

3 $\mathcal{A}$-satisfiability versus $\mathcal{M}$-satisfiability

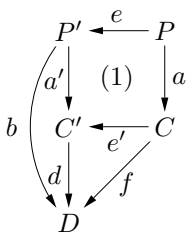
In this section, we investigate the different satisfiability notions. One may ask whether $\mathcal{A}$-satisfiability and $\mathcal{M}$-satisfiability are expressively equivalent.

Assumption. We assume that $\langle \mathcal{C}, \mathcal{M} \rangle$ is a weak adhesive HLR category with epi-$\mathcal{M}$-factorizations, that is, for every morphism there is an epi-mono-factorization with monomorphism in $\mathcal{M}$.

There is a transformation from $\mathcal{A}$- to $\mathcal{M}$-satisfiability for morphisms. The construction is a quotient construction on conditions and similar to the transformation of constraints into application conditions in [3, 8].

Theorem 1 (for morphisms: from $\mathcal{A}$- to $\mathcal{M}$-satisfiability). *There is a transformation* M *on conditions such that, for every condition c over P and every morphism $p\colon P \to G$, $p \models \mathrm{M}(c) \Leftrightarrow p \models_{\mathcal{A}} c$.*

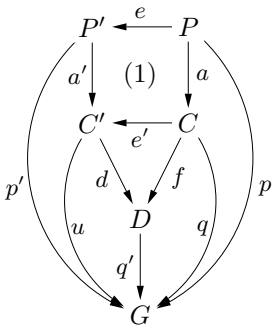
Construction. Let $\mathrm{M}(c) = \vee_{e \in \mathcal{E}} \exists(e, \mathrm{M}_e(c))$ where the disjunction $\vee_e$ ranges over all epimorphisms $e\colon P \to P'$ and, for every epimorphism $e\colon P \to P'$, the transformation M_e is defined inductively on the structure of the conditions:



$$\begin{aligned} \mathrm{M}_e(\exists a) &= \vee_d \exists b \\ \mathrm{M}_e(\exists(a, c)) &= \vee_d \exists(b, \mathrm{M}_f(c)) \end{aligned}$$

where (1) is the pushout of the morphisms $a\colon P \to C$ and $e\colon P \to P'$ leading to morphisms $a'\colon P' \to C'$ and $e'\colon C \to C'$ and the disjunction $\vee_d$ ranges over all epimorphisms $d\colon C' \to D$ such that $b = d \circ a'$ is in $\mathcal{M}$ and $f = d \circ e'$. For Boolean conditions, the transformations are extended in the usual way.

Proof. By structural induction, we show: For every condition c over P and every morphism $p\colon P \to G$, $p' \models \mathrm{M}_e(c) \Leftrightarrow p \models_{\mathcal{A}} c$ for some epi-$\mathcal{M}$-factorization $p = p' \circ e$ with epimorphism $e\colon P \to P'$ and monomorphism $p'\colon P' \to G$ in $\mathcal{M}$. For conditions of the form $\exists(a, c)$ the statement is proved as follows: **If.** Let $p' \models \mathrm{M}_f(\exists(a, c))$. Then there is some epimorphism $d\colon C' \to D$ with $b = d \circ a'$ in $\mathcal{M}$ and $f = d \circ e'$ such that $p' \models \exists(b, \mathrm{M}_f(c))$. By definition of $\mathcal{M}$-satisfiability, there is some $q'\colon D \to G$ in $\mathcal{M}$ such that $q' \circ b = p'$ and $q' \models \mathrm{M}_e(c)$. Define $q = q' \circ f$. Then $q \circ a = p$ and, by inductive hypothesis, $q \models_{\mathcal{A}} c$. Consequently, $p \models_{\mathcal{A}} \exists(a, c)$.



Only if. Let $p \models_{\mathcal{A}} \exists(a, c)$. Then there is some $q\colon C \to G$ such that $q \circ a = p$ and $q \models_{\mathcal{A}} c$. By the universal property of pushouts, there is some $u\colon C' \to G$ with $u \circ a' = p'$ and $u \circ e' = q$. Let $u = q' \circ d$ be an epi-$\mathcal{M}$-factorization of u with epimorphism d and monomorphism q' in $\mathcal{M}$. Then $q' \circ b = p'$. Since $\mathcal{M}$ is closed under decompositions, p' and q' in $\mathcal{M}$ imply b in $\mathcal{M}$. By inductive hypothesis, $q' \models \mathrm{M}_f(c)$. Consequently, $p' \models \vee_d \exists(b, \mathrm{M}_f(c)) = \mathrm{M}_e(\exists(a, c))$. For conditions of the form $\exists a$, the proof is similar. For Boolean conditions, the statement follows from the definitions and the inductive hypothesis. Consequently, the statement holds for all conditions.

The statement implies that, for every morphism $p\colon P \to G$, $p \models \mathrm{M}(c)$ iff $p' \models \mathrm{M}_e(c)$ for some epi-$\mathcal{M}$-factorization $p = p' \circ e$ of p with epimorphism e and monomorphism p' in $\mathcal{M}$. This completes the proof.

Remark 2. The transformation M in Theorem 1 on conditions is very similar to the transformation of constraints into application conditions in [3, 8] and the original paper of [10].

Example 4. In the category of graphs, the condition $c = \neg\exists(\underset{1}{\bigcirc} \to \underset{1}{\bigcirc\!\!\rightarrow\!\!\bigcirc})$ with meaning for graph morphisms w.r.t. $\mathcal{A}$-satisfiability "There does not exist an outgoing edge" is transformed into the condition $\mathrm{M}(c)$ meaning w.r.t. $\mathcal{M}$-satisfiability "There does not exist a proper outgoing edge or a loop".

$$\begin{aligned}\mathrm{M}(c) &= \mathrm{M}(\neg\exists(\underset{1}{\bigcirc} \to \underset{1}{\bigcirc\!\!\rightarrow\!\!\bigcirc})) = \neg\mathrm{M}(\exists(\underset{1}{\bigcirc} \to \underset{1}{\bigcirc\!\!\rightarrow\!\!\bigcirc})) = \neg \vee_{e\in\mathcal{E}} \exists(e, \mathrm{M}_e(c)) \\ &= \neg\exists(\mathrm{id}, \mathrm{M}_{\mathrm{id}}(c)) = \neg\exists(\mathrm{id}, \exists(\underset{1}{\bigcirc} \to \underset{1}{\bigcirc\!\!\rightarrow\!\!\bigcirc}) \vee \exists(\bigcirc \to \bigcirc)) \\ &\equiv \neg(\exists(\underset{1}{\bigcirc} \to \underset{1}{\bigcirc\!\!\rightarrow\!\!\bigcirc}) \vee \exists(\bigcirc \to \bigcirc))\end{aligned}$$

As corollary, we obtain a transformation from $\mathcal{A}$- to $\mathcal{M}$-satisfiability for objects.

Corollary 1 (for objects: from $\mathcal{A}$- to $\mathcal{M}$-satisfiability). There is a transformation M^* on conditions such that, for every condition c over P and every object G, $G \models \mathrm{M}^*(c) \Leftrightarrow G \models_{\mathcal{A}} c$.

Proof. Let $\mathrm{M}^*(c) = \wedge_{e\in\mathcal{E}} \mathrm{M}_e(c)$ where $\mathcal{E}$ denotes the set of all epimorphisms starting from P and $\mathrm{M}_e(c)$ is as in the construction of Theorem 1. By the definition of $\mathcal{A}$-satisfiability of objects, epi-$\mathcal{M}$-factorization of morphisms, the proof of

Theorem 1, and the definition of $\mathcal{M}$-satisfiability of objects, we have $G \models_{\mathcal{A}} c$ iff for all $p\colon P \to G$, $p \models_{\mathcal{A}} c$ iff for all $p\colon P \to G$ with epi-$\mathcal{M}$-factorization $p = p' \circ e$, $p' \models \mathrm{M}_e(c)$ iff for all $p' \in \mathcal{M}$, $p' \models \wedge_{e \in \mathcal{E}} \mathrm{M}_e(c)$ iff $G \models \mathrm{M}^*(c)$.

Example 5. The condition $c = \exists($  $)$ with meaning for graphs w.r.t. $\mathcal{A}$-satisfiability "For every node, there exists an outgoing edge" is transformed into the condition $\mathrm{M}^*(c) = \exists($  $) \vee \exists($  $)$ meaning w.r.t. $\mathcal{M}$-satisfiability "For every node, there exists a proper outgoing edge or a loop". The condition $c = \exists($  $)$ with meaning for graphs w.r.t. $\mathcal{A}$-satisfiability "For every pair of nodes, there exists a connecting edge" is transformed into the condition $\mathrm{M}^*(c) = \exists($  $) \wedge \exists($  $)$ meaning w.r.t. $\mathcal{M}$-satisfiability "For every pair of distinct nodes, there exists a connecting edge and, for every node, there exists a loop".

There is a transformation from $\mathcal{M}$- to $\mathcal{A}$-satisfiability for morphisms, provided that $\mathcal{M}$ is strictly closed under decompositions, i.e. $g \circ f \in \mathcal{M}$ implies $f \in \mathcal{M}$.

Theorem 2 (for morphisms: from $\mathcal{M}$- to $\mathcal{A}$-satisfiability). *Let $\mathcal{M}$ be strictly closed under decompositions. Then there is a transformation* A *on conditions such that, for every condition c over an object P and every morphism $p\colon P \to G$, $p \models_{\mathcal{A}} \mathrm{A}(c) \Leftrightarrow p \models c$.*

Construction. The transformation A is defined inductively on the structure of the conditions: For a morphism $a\colon P \to C$ and a condition c over C,

$$\begin{aligned} \mathrm{A}(\exists a) &= \exists(a, \mathrm{inM}_C) \\ \mathrm{A}(\exists(a, c)) &= \exists(a, \mathrm{inM}_C \wedge \mathrm{A}(c)) \end{aligned}$$

where $\mathrm{inM}_C = \neg \vee_e \exists e$ is a condition over C and the disjunction $\vee_e$ ranges over all epimorphisms $e\colon C \to C'$ not in $\mathcal{M}$. For Boolean conditions, the transformation is as usual.

Proof. The property "the match is in $\mathcal{M}$" can be expressed by an application condition inM_C: For every morphism $q\colon C \to G$, $q \in \mathcal{M} \Leftrightarrow q \models_{\mathcal{A}} \mathrm{inM}_C$. This may be seen as follows: If $q \not\models_{\mathcal{A}} \mathrm{inM}_C$, then $q \models_{\mathcal{A}} \exists e$ for some epimorphism $e\colon C \to C'$ not in $\mathcal{M}$. Then there is some $q'\colon C' \to G$ such that $q' \circ e = q$. Then q is not in $\mathcal{M}$. Otherwise, by the strict closure of $\mathcal{M}$ under decompositions, q in $\mathcal{M}$ would imply e in $\mathcal{M}$. If q is not in $\mathcal{M}$, we consider an epi-$\mathcal{M}$ factorization $q = q' \circ e$ of q with epimorphism e and monomorphism q' in $\mathcal{M}$. Then e is not in $\mathcal{M}$. Otherwise, by closure of $\mathcal{M}$ under compositions, e and q' in $\mathcal{M}$ would imply q in $\mathcal{M}$.
By structural induction, we show the statement of the theorem. For conditions of the form $\exists(a, c)$ we have the following. **Only if**. Let $p \models \exists(a, c)$. Then there is a morphism $q\colon C \to G$ in $\mathcal{M}$ such that $q \circ a = p$ and $q \models c$. By the inductive hypothesis and the application condition inM_C being equivalent to "match is in M", $q \models_{\mathcal{A}} \mathrm{inM}_C$ and $q \models_{\mathcal{A}} \mathrm{A}(c)$. Consequently, $p \models_{\mathcal{A}} \exists(a, \mathrm{inM}_C \wedge \mathrm{A}(c)) =$

$\mathrm{A}(\exists(a,c))$. **If.** Let $p \models_{\mathcal{A}} \mathrm{A}(\exists(a,c))$. Then there is some $q\colon C \to G$ such that $q \circ a = p$, $q \models_{\mathcal{A}} \mathrm{inM}_C$, and $q \models_{\mathcal{A}} \mathrm{A}(c)$. By the inductive hypothesis, $q \in \mathcal{M}$ and $q \models c$. Thus, $p \models \exists(a,c)$. For conditions of the form $\exists a$ the proof is similar. For Boolean conditions, the statement follows from the definitions and the inductive hypothesis. This completes the inductive proof.

Example 6. The class $\mathcal{M}$ of injective graph morphisms is strictly closed under decompositions. The condition $c = \exists(\bigcirc\,\bigcirc \to \bigcirc\rightarrowtail\bigcirc)$ with meaning for graph morphisms w.r.t. $\mathcal{M}$-satisfiability "For the two nodes, there exists a connecting edge" is transformed into the condition $\mathrm{A}(c) = \exists(\bigcirc\,\bigcirc \to \bigcirc\rightarrowtail\bigcirc, \neg\exists(\bigcirc\rightarrowtail\bigcirc \to \circlearrowright))$ meaning w.r.t. $\mathcal{A}$-satisfiability "For the nodes, there exists a connecting edge and the endpoints are distinct".

There is a transformation from $\mathcal{M}$- to $\mathcal{A}$-satisfiability for objects, provided that $\mathcal{M}$ is strictly closed under decompositions and the category has an $\mathcal{M}$-initial object: In a category $\mathcal{C}$, an object I is *initial*, if for every object G in $\mathcal{C}$, there exists a unique initial morphism $i\colon I \to G$. In a weak adhesive category $\langle \mathcal{C}, \mathcal{M} \rangle$, an object I is $\mathcal{M}$*-initial* if I is initial in $\mathcal{C}$ and the initial morphisms are in $\mathcal{M}$.

Corollary 2 (for objects: from $\mathcal{M}$- to $\mathcal{A}$-satisfiability). For weak adhesive HLR categories with $\mathcal{M}$-initial object and $\mathcal{M}$ strictly closed under decompositions, there is a transformation A^* on conditions such that, for every condition c over an object P and every object G, $G \models_{\mathcal{A}} \mathrm{A}^*(c) \Leftrightarrow G \models c$.

Proof. Let $\mathrm{A}^*(c) = \forall(i, \mathrm{inM}_P \Rightarrow \mathrm{A}(c))$ where $\forall(a,d)$ abbreviates the condition $\neg\exists(a, \neg d)$ and i denotes the unique morphism from the initial object I to P in $\mathcal{M}$. By the definition of $\mathcal{M}$-satisfiability, Theorem 2, the property of inM_P, and the definition of $\mathcal{A}$-satisfiability for objects, we have $G \models c$ iff for all $p\colon P \to G$ in $\mathcal{M}$, $p \models c$ iff for all $p\colon P \to G$, $p \models_{\mathcal{A}} \mathrm{inM}_P$ implies $p \models_{\mathcal{A}} \mathrm{A}(c)$ iff for all $p\colon P \to G$, $p \models_{\mathcal{A}} \mathrm{inM}_P \Rightarrow \mathrm{A}(c)$ iff for $j\colon I \to G$ in $\mathcal{M}$, $j \models_{\mathcal{A}} \forall(i, \mathrm{inM}_P \Rightarrow \mathrm{A}(c))$ iff $G \models_{\mathcal{A}} \forall(i, \mathrm{inM}_P \Rightarrow \mathrm{A}(c))$ iff $G \models_{\mathcal{A}} \mathrm{A}^*(c)$.

Example 7. The category $\langle \mathbf{Graphs}, \mathcal{M} \rangle$ has an $\mathcal{M}$-initial object: the empty graph, denoted by $\emptyset$. The condition $c = \exists(\bigcirc\,\bigcirc \to \bigcirc\rightarrowtail\bigcirc)$ with the meaning for graph morphisms w.r.t. $\mathcal{M}$-satisfiability "Every pair of distinct nodes is connected by an edge" is transformed into the condition

$$\mathrm{A}^*(c) = \forall(\emptyset \to \bigcirc\,\bigcirc, \neg\exists(\bigcirc\,\bigcirc \to \bigcirc) \Rightarrow \exists(\bigcirc\,\bigcirc \to \bigcirc\rightarrowtail\bigcirc, \neg\exists(\bigcirc\rightarrowtail\bigcirc \to \circlearrowright))$$

meaning w.r.t. $\mathcal{A}$-satisfiability "For every pair of nodes, whenever the nodes are distinct, then there is a connecting edge (and the endpoints are distinct)".

By Theorems 1 and 2 and Corollaries 1 and 2 we obtain the following corollary.

Corollary 3. For weak adhesive HLR categories with epi-$\mathcal{M}$-factorizations, $\mathcal{M}$-initial object, and $\mathcal{M}$ strictly closed under decompositions, $\mathcal{A}$-satisfiability and $\mathcal{M}$-satisfiability are expressively equivalent.

Remark 3. The equivalence result is valid for nested constraints and application conditions [8]; it is not valid for plain constraints in the sense of [3].

Finally, we present two transformations of conditions over morphisms, i.e. given a morphism $e\colon P \to P'$, then the transformation transforms conditions over P into conditions over P'.

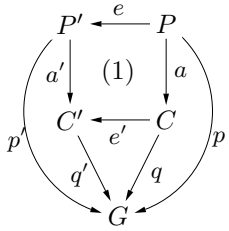
Lemma 1 (from $\mathcal{A}$- to $\mathcal{A}$-satisfiability). *For every (epi)morphism $e\colon P \to P'$, there is a transformation AA_e such that, for every condition c over P and every morphism $p'\colon P' \to G$, $p' \models_{\mathcal{A}} \mathrm{AA}_e(c) \Leftrightarrow p' \circ e \models_{\mathcal{A}} c$.*

Construction. For a morphism $e\colon P \to P'$, the transformation AA_e is defined inductively on the structure of the conditions:

$$\begin{aligned} \mathrm{AA}_e(\exists a) &= \exists a' \\ \mathrm{AA}_e(\exists(a,c)) &= \exists(a', \mathrm{AA}_{e'}(c)) \end{aligned}$$

where (1) is the pushout of the morphisms a and e leading to the morphisms $a'\colon P' \to C'$ and $e'\colon C \to C'$. For Boolean conditions, the transformation is extended in the usual way.

Proof. By structural induction. For conditions of the form $\exists(a,c)$, the statement is proved as follows: **If.** Let $p' \models_{\mathcal{A}} \mathrm{AA}_e(\exists(a,c))$. Then there is some $q'\colon C' \to G$ such that $q' \circ a' = p'$ and $q' \models_{\mathcal{A}} \mathrm{AA}_{e'}(c)$. By inductive hypothesis, $q' \circ e' \models_{\mathcal{A}} c$ Then $q' \circ e' \circ a = p' \circ e$. Consequently, $p' \circ e \models_{\mathcal{A}} \exists(a,c)$.



Only if. Let $p' \circ e \models_{\mathcal{A}} \exists(a,c)$. Then there is some $q\colon C \to G$ such that $q \circ a = p' \circ e$ and $q \models_{\mathcal{A}} c$. By the universal property of pushouts, there is a morphism $q'\colon C' \to G$ such that $q' \circ a' = p'$ and $q' \circ e' = q$. By inductive hypothesis, $q' \models_{\mathcal{A}} \mathrm{AA}_{e'}(c)$. Consequently, $p' \models_{\mathcal{A}} \exists(a', \mathrm{AA}_{e'}(c)) = \mathrm{AA}_e(\exists(a,c))$. For conditions of the form $\exists a$, the proof is similar.

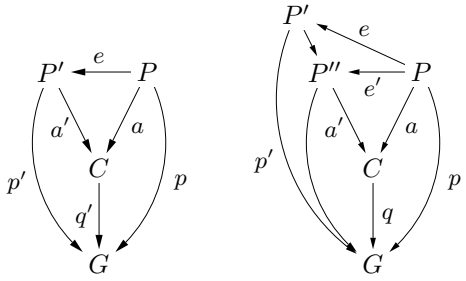
For Boolean conditions, the statement follows from the definitions and the inductive hypothesis. Consequently, the statement holds for all conditions.

Example 8. Given the condition $c = \neg\exists(\bigcirc\,\bigcirc \to \bigcirc\rightarrowtail\bigcirc)$ with the meaning for graph morphisms w.r.t. $\mathcal{A}$-satisfiability "There does not exist a connecting edge" and the epimorphism $e\colon \bigcirc\,\bigcirc \to \bigcirc$, $\mathrm{AA}_e(c) = \neg\exists(\bigcirc \to \circlearrowright)$ is the condition with the meaning "There does not exist an attached loop".

Lemma 2 (from $\mathcal{M}$- to $\mathcal{M}$-satisfiability). *For every epimorphism $e\colon P \to P'$, there is a transformation MM_e such that, for every condition c over P and every morphism $p'\colon P' \to G$ in $\mathcal{M}$, $p' \models \mathrm{MM}_e(c) \Leftrightarrow p' \circ e \models c$.*

Construction. For every epimorphism e, the transformation MM_e is defined inductively on the structure of the conditions: $\mathrm{MM}_e(\exists a) = \exists a'$ and $\mathrm{MM}_e(\exists(a,c)) = \exists(a',c)$ if e is the epimorphism of an epi-$\mathcal{M}$-factorization $e \circ a' = a$ and $\mathrm{MM}_e(\exists a) = \mathrm{MM}_e(\exists(a,c)) = \text{false}$ otherwise. For Boolean conditions, the transformation is extended in the usual way.

Proof. By structural induction. For conditions of the form $\exists(a,c)$, the statement is proved as follows: **If.** Let $p' \models \exists(a',c)$ for some epi-$\mathcal{M}$-factorization $a = a' \circ e$ of a. Then there is some $q' \colon C \to G$ in $\mathcal{M}$ such that $q' \circ a' = p'$ and $q' \models c$. Moreover, $q' \circ a = p' \circ e$. Consequently, $p' \circ e \models \exists(a,c)$. If e is not the epimorphism of an epi-$\mathcal{M}$-factorization of a, we have $p' \not\models \mathrm{MM}_e(\exists(a,c)) = \text{false}$ and $p' \circ e \not\models \exists(a,c)$.



Only if. Let $p' \circ e \models \exists(a,c)$. Then there is some morphism $q \colon C \to G$ in $\mathcal{M}$ such that $q \circ a = p$ and $q \models c$ and some morphism $q' \colon P'' \to G$ in $\mathcal{M}$ with $q' = q \circ a'$. Whenever $a' \circ e'$ is an epi-$\mathcal{M}$-factorization of a, $q' \circ e'$ is an epi-$\mathcal{M}$-factorization of $p = p' \circ e$. By the uniqueness of epi-$\mathcal{M}$-factorizations, we have $e = e'$ and $p' = q'$ (up to isomorphism). Then $p' = q' \models \exists(a',c)$. For conditions of the form $\exists a$, the proof is similar. For Boolean conditions, the statement follows from the definitions and the inductive hypothesis. Consequently, the statement holds for all conditions.

Example 9. The condition $c = \exists($ ○ ○ →  $)$ with the meaning for graph morphisms w.r.t. $\mathcal{M}$-satisfiability "One of the nodes is connected by a loop" is transformed over the surjective graph morphism $e \colon$ ○ ○ → ○ into the condition $\mathrm{MM}_e(c) = \exists($ ○ →  $)$ with the meaning "The node has an attached loop". The condition $c' = \exists($ ○ ○ → ○→○ $)$ with the meaning for graph morphisms w.r.t. $\mathcal{M}$-satisfiability "For the two nodes, there exists a connecting edge" is transformed into the condition $\mathrm{MM}_e(c') = \text{false}$ with the meaning "Not satisfiable".

By Lemmas 1 and 2 and Theorems 1 and 2 we obtain the following corollary.

Corollary 4 (from $\mathcal{A}$- to $\mathcal{M}$ and $\mathcal{M}$- to $\mathcal{A}$-satisfiability). For every epimorphism $e \colon P \to P'$, there are transformations AM_e and MA_e such that, for every condition c over P and every morphism $p' \colon P' \to G$ in $\mathcal{M}$, $p' \models \mathrm{AM}_e(c) \Leftrightarrow p' \circ e \models_{\mathcal{A}} c$ and $p' \models_{\mathcal{A}} \mathrm{MA}_e(c) \Leftrightarrow p' \circ e \models c$.

Construction. For every epimorphism $e \colon P \to P'$, let $\mathrm{AM}_e = \mathrm{M} \circ \mathrm{AA}_e$ and $\mathrm{MA}_e = \mathrm{A} \circ \mathrm{MM}_e$.

4 $\mathcal{A}$-matching versus $\mathcal{M}$-matching

The definition of a direct derivation allows $\mathcal{A}$-matching as well as $\mathcal{M}$-matching. One may ask whether $\mathcal{A}$-matching and $\mathcal{M}$-matching are expressively equivalent. We establish a Simulation Theorem saying that any direct derivation with arbitrary matching can be simulated by a direct derivation with $\mathcal{M}$-matching. This extends the Simulation Theorem in [7] to weak adhesive HLR categories with epi-$\mathcal{M}$-factorizations and makes use of so-called quotient rules and epi-$\mathcal{M}$-factorizations of morphisms and direct derivations.

Recall that a direct derivation $G \Rightarrow_{\langle q,\mathrm{ac}\rangle,m,m^*} H$ in $\langle X, Y\rangle$ with X, Y in $\{\mathcal{A}, \mathcal{M}\}$ is a direct derivation $G \Rightarrow_{q,m,m^*} H$ with $m \in X$, $m \models_Y \mathrm{ac}_L$, and $m^* \models_Y \mathrm{ac}_R$.

Theorem 3 (from $\mathcal{A}$- to $\mathcal{M}$-matching). *For every $Y \in \{\mathcal{A}, \mathcal{M}\}$, there is a transformation Q_Y from rules into sets of rules such that, for every rule p,*

$$G \Rightarrow_p H \text{ in } \langle \mathcal{A}, Y\rangle \text{ if and only if } G \Rightarrow_{\mathrm{Q}_Y(p)} H \text{ in } \langle \mathcal{M}, Y\rangle.$$

Construction. For a rule $q = \langle L \leftarrow K \rightarrow R\rangle$, the rule $q' = \langle L' \leftarrow K' \rightarrow R'\rangle$ is a *quotient rule* of q if there are two pushouts of the form

$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ {\scriptstyle e}\downarrow & (1) & \downarrow & (2) & \downarrow{\scriptstyle e^*} \\ L' & \longleftarrow & K' & \longrightarrow & R' \end{array}$$

where the vertical morphisms are epimorphisms. For a rule $p = \langle q, \mathrm{ac}\rangle$ with application condition, $\mathrm{Q}_{\mathcal{A}}(p)$ is the set of rules $p' = \langle q', \mathrm{ac}'\rangle$ where q' is a quotient rule of q, $\mathrm{ac}'_L = \mathrm{AA}_e(\mathrm{ac}_L)$ and $\mathrm{ac}'_R = \mathrm{AA}_{e^*}(\mathrm{ac}_R)$. $\mathrm{Q}_{\mathcal{M}}(p)$ is the set of rules $p' = \langle q', \mathrm{ac}'\rangle$ where q' is a quotient rule of q, $\mathrm{ac}'_L = \mathrm{MM}_e(\mathrm{ac}_L)$, and $\mathrm{ac}'_R = \mathrm{MM}_{e^*}(\mathrm{ac}_R)$.

Proof. Let $G \Rightarrow_{p',n,n^*} H$ be a direct derivation in $\langle \mathcal{M}, Y\rangle$ through $p' = \langle q', \mathrm{ac}'\rangle$ in $\mathrm{Q}_Y(p)$ with $q' = \langle L' \leftarrow K' \rightarrow R'\rangle$ and ac' as in the construction. Then the diagrams (1), (2), (1') and (2') in the figure below are pushouts. By the Composition Lemma of pushouts valid in every category, the composed diagrams (1)+(1') and (2)+(2') are pushouts as well. Hence, there is a direct derivation $G \Rightarrow_{q,m,m^*} H$ in $\langle \mathcal{A}, Y\rangle$. By assumption, $n \models_Y \mathrm{ac}'_L$ and $n^* \models_Y \mathrm{ac}'_R$. By Lemma 1 [2], $m \models_Y \mathrm{ac}_L$ and $m^* \models_Y \mathrm{ac}_R$. Thus, $G \Rightarrow_{p,m,m^*} H$ is a direct derivation in $\langle \mathcal{A}, Y\rangle$.

$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ {\scriptstyle e}\downarrow & (1) & \downarrow & (2) & \downarrow{\scriptstyle e^*} \\ L' & \longleftarrow & K' & \longrightarrow & R' \\ {\scriptstyle n}\downarrow & (1') & \downarrow & (2') & \downarrow{\scriptstyle n^*} \\ G & \longleftarrow & D & \longrightarrow & H \end{array} \qquad m \text{ (} L \to L' \text{)}, \quad m^* \text{ (} R \to R' \text{)}$$

Vice versa, let $G \Rightarrow_{p,m,m^*} H$ be a direct derivation in $\langle \mathcal{A}, Y \rangle$ through $p = \langle q, \mathrm{ac} \rangle$ with $q = \langle L \leftarrow K \rightarrow R \rangle$. Let $m = n \circ e$ an epi-$\mathcal{M}$ factorization of m with epimorphism $e\colon L \rightarrow L'$ and monomorphism $n\colon L' \rightarrow G$ in $\mathcal{M}$. Then there is a decomposition of the original diagrams into diagrams (1) and (1'), (2) and (2') as follows: Construct K' as a pullback object of $L' \rightarrow G \leftarrow D$ and denote the diagram by (1'). By the universal property of pullbacks, there is a unique morphism $e'\colon K \rightarrow K'$ such that $K \rightarrow K' \rightarrow D = K \rightarrow D$ and diagram (1) commutes. By the pushout-pullback decomposition, (1') and (1) are pushouts. Now construct R' as the pushout object of $K' \leftarrow K \rightarrow R$ and denote the diagram by (2). By the universal property of pushouts, there is a unique morphism $n^*\colon R' \rightarrow H$ such that $R \rightarrow R' \rightarrow H = R \rightarrow H$ and diagram (2') commutes. By the Decomposition Lemma for pushouts valid in every category, diagram (2') is a pushout. Since epimorphisms and $\mathcal{M}$-morphisms are closed under pullbacks and pushouts, the vertical morphisms e, e', and e^* are epimorphisms and the vertical morphisms n, n', and n^* are in $\mathcal{M}$. Then $q' = \langle L' \leftarrow K' \rightarrow R' \rangle$ is a quotient rule of q and $p' = \langle q', \mathrm{ac}' \rangle$ with $\mathrm{ac}' = \langle \mathrm{AA}_e(\mathrm{ac}_L), \mathrm{AA}_{e^*}(\mathrm{ac}_R) \rangle$ $[\langle \mathrm{MM}_e(\mathrm{ac}_L), \mathrm{MM}_{e^*}(\mathrm{ac}_R) \rangle]$ is a quotient rule or p in $\mathrm{Q}_{\mathcal{A}}(p)$ $[\mathrm{Q}_{\mathcal{M}}(p)]$. By Lemma 1 [2], $G \Rightarrow_{p',n,n^*} H$ is a direct derivation in $\langle \mathcal{M}, \mathcal{A} \rangle$ $[\langle \mathcal{M}, \mathcal{M} \rangle]$. This completes the proof.

Remark 4. The construction and proof for the transformation Q of rules in Theorem 3 is very similar to transformation of right- to left application conditions in [3, 8].

Example 10. Consider the rule $p = \langle q, \mathrm{ac} \rangle$ with $q = \langle \bigcirc\,\bigcirc \leftarrow \bigcirc\,\bigcirc \rightarrow \bigcirc\!\!\rightarrowtail\!\!\bigcirc \rangle$ and $\mathrm{ac}_L = \neg\exists(\bigcirc\,\bigcirc \rightarrow \bigcirc\!\!\rightarrowtail\!\!\bigcirc)$ with $\langle \mathcal{A}, Y \rangle$-meaning "Add a connecting edge, provided there does not exist one". Then $p' = \langle q', \mathrm{ac}'_L) \rangle$ with $q' = \langle \bigcirc \leftarrow \bigcirc \rightarrow \circlearrowright \rangle$ and $\mathrm{ac}'_L = \neg\exists(\bigcirc \rightarrow \circlearrowright)$ is a rule in $\mathrm{Q}_{\mathcal{A}}(p)$ with $\langle \mathcal{M}, \mathcal{A} \rangle$-meaning "Add a loop at the node, provided there does not exist one". Furthermore, $p'' = \langle q', \mathrm{ac}''_L \rangle$ with $\mathrm{ac}''_L = \mathrm{false}$ is a rule in $\mathrm{Q}_{\mathcal{M}}(p)$ with $\langle \mathcal{M}, \mathcal{M} \rangle$-meaning "Never add a loop at the node".

We can simulate $\mathcal{M}$-matching by $\mathcal{A}$-matching by using the application condition inM which is satisfied iff the match is in $\mathcal{M}$.

Theorem 4 (from $\mathcal{M}$- to $\mathcal{A}$-matching). *Let $\mathcal{M}$ be strictly closed under decompositions and $Y \in \{\mathcal{A}, \mathcal{M}\}$. Then there is a transformation* R *on rules such that, for every rule p, $G \Rightarrow_p H$ in $\langle \mathcal{M}, Y \rangle$ iff $G \Rightarrow_{\mathrm{R}(p)} H$ in $\langle \mathcal{A}, Y \rangle$.*

Construction. For every rule $p = \langle q, \mathrm{ac} \rangle$, let $\mathrm{R}(p) = \langle q, \mathrm{ac}' \rangle$ with

$$\mathrm{ac}' = \langle \mathrm{inM}_L \wedge \mathrm{ac}_L, \mathrm{inM}_R \wedge \mathrm{ac}_R \rangle.$$

Proof. By the proof of Theorem 2, the property "the match is in $\mathcal{M}$" can be expressed by the application condition inM. Thus, for every rule p, $G \Rightarrow_p H$ in $\langle \mathcal{M}, Y \rangle$ if and only if $G \Rightarrow_{\mathrm{R}(p)} H$ in $\langle \mathcal{A}, Y \rangle$.

Example 11. Consider the rule $p = \langle q, \mathrm{ac}\rangle$ with $q = \langle \bigcirc\,\bigcirc \leftarrow \bigcirc\,\bigcirc \rightarrow \bigcirc\!\!\rightarrowtail\!\!\bigcirc\rangle$ and $\mathrm{ac}_L = \neg\exists(\bigcirc\,\bigcirc \rightarrow \bigcirc\!\!\rightarrowtail\!\!\bigcirc)$ with $\langle\mathcal{M}, Y\rangle$-meaning "Add an edge between distinct nodes, provided there does not exist a connecting edge". Then $\mathrm{R}(p) = \langle q, \neg\exists(\bigcirc\,\bigcirc \rightarrow \bigcirc) \wedge \mathrm{ac}_L\rangle$ is a rule with $\langle\mathcal{A}, Y\rangle$-meaning "Add an edge between the nodes, provided the nodes are distinct and there does not exist a connecting edge".

As a consequence, we obtain transformations for all possible kinds of direct derivations.

Corollary 5. For weak adhesive HLR categories with epi-$\mathcal{M}$-factorizations and $\mathcal{M}$ strictly closed under decompositions and tuples $t = \langle X, Y, X', Y'\rangle \in \{\mathcal{A}, \mathcal{M}\}^4$, there is a transformation Q_t such that, for every rule p,

$$G \Rightarrow_p H \text{ in } \langle X, Y\rangle \text{ if and only if } G \Rightarrow_{\mathrm{Q}_t(p)} H \text{ in } \langle X', Y'\rangle.$$

Proof. The transformation results follow directly from the transformation results on matching and satisfiability.

Finally, we obtain the following corollary.

Corollary 6. For weak adhesive HLR categories with epi-$\mathcal{M}$-factorizations and $\mathcal{M}$ strictly closed under decompositions, $\mathcal{A}$-matching and $\mathcal{M}$-matching are expressively equivalent.

5 Conclusion

We have shown that all notions of matching and satisfiability have their advantages and disadvantages and, for weak adhesive HLR categories with epi-$\mathcal{M}$-factorizations, $\mathcal{M}$-initial object, and $\mathcal{M}$ strictly closed under decompositions,

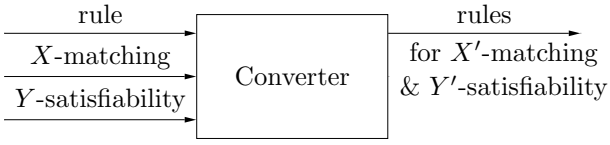
- $\mathcal{A}$-matching and $\mathcal{M}$-matching
- $\mathcal{A}$-satisfiability and $\mathcal{M}$-satisfiability

are expressively equivalent. The equivalence results are valid for nested constraints and application conditions in the sense of [8]; they are not valid for (basic) constraints in the sense of [3]. We have presented some transformation results for application conditions, plain rules, and rules with application conditions.

	For application conditions:	
M	from $\mathcal{A}$- to $\mathcal{M}$-satisfiability	Theorem 1
A	from $\mathcal{M}$- to $\mathcal{A}$-satisfiability	Theorem 2
AA_e	from $\mathcal{A}$- to $\mathcal{A}$-satisfiability	Lemma 1
AM_e	from $\mathcal{A}$- to $\mathcal{M}$-satisfiability	Corollary 4
MA_e	from $\mathcal{M}$- to $\mathcal{A}$-satisfiability	Corollary 4
MM_e	from $\mathcal{M}$- to $\mathcal{M}$-satisfiability	Lemma 2

	For plain rules:	
Q	from $\mathcal{A}$- to $\mathcal{M}$-matching	Theorem 3
R	from $\mathcal{M}$- to $\mathcal{A}$-matching	Theorem 4
	For rules:	
	from $\langle X, Y \rangle$ to $\langle X', Y' \rangle$	Corollary 5

This allows to switch from every combination of X-matching and Y-satisfiability to every other combination of X'-matching and Y'-satisfiability and may be the basis of a converter.



Summarizing, $\mathcal{A}$-matching as well as $\mathcal{M}$-matching are adequate matching notions and $\mathcal{A}$-satisfiability as well as $\mathcal{M}$-satisfiability are adequate satisfiability notions. Nevertheless, we propose to consider either $\mathcal{A}$-matching and $\mathcal{A}$-satisfiability or $\mathcal{M}$-matching and $\mathcal{M}$-satisfiability.

- The combination $\mathcal{A}$-matching and $\mathcal{M}$-satisfiability does not fit well together: Conditions of the form $c = \exists a$ or $\exists(a, c)$ are only satisfiable by morphisms with a certain degree of identifaction that depends on a. In fact, if a is an $\mathcal{M}$-morphism, $m \models c$ implies m in $\mathcal{M}$.
- In the case of $\mathcal{A}$-matching and $\mathcal{A}$-satisfiability, the transformation from constraints to application conditions is more simple and natural than the one for $\mathcal{A}$-matching and $\mathcal{M}$-satisfiability. The same holds for a transformation from application conditions to constraints. However, no direct transformation from right into left application conditions (L) is known, just the junction of A $\circ$ L $\circ$ M requiring that the category has an $\mathcal{M}$-initial object and $\mathcal{M}$ is strictly closed under decomposition.
- In the case of $\mathcal{M}$-matching and $\mathcal{M}$-satisfiability, the transformations from constraints to application conditions, from application conditions to constraints, and, consequently, the construction of weakest preconditions for high-level programs [9], are simpler and more intuitive than the one of $\mathcal{A}$-matching and $\mathcal{M}$-satisfiability.

Considering the above and the fact that $\mathcal{M}$-matching is the more explicit matching notion, our choice is $\mathcal{M}$-matching and $\mathcal{M}$-satisfiability.

Further topics will be the followings.

(1) Comparison of notions: A comparison of conditions – as considered in this paper – and first-order formulas on graphs and high-level structures.
(2) Extensions of the theory: The investigation of weak adhesive high-level replacement systems with merging similar to the investigation of graph replacement systems with merging as in [7].

(3) Implementation: A system for converting conditions and rules with one kind of matching and satisfiability into conditions and rules with the implemented kind of matching and satisfiability.

References

1. A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, and M. Löwe. Algebraic approaches to graph transformation. Part I: Basic concepts and double pushout approach. In *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1, pages 163–245. World Scientific, 1997.
2. H. Ehrig. Introduction to the algebraic theory of graph grammars. In *Graph-Grammars and Their Application to Computer Science and Biology*, volume 73 of *Lecture Notes in Computer Science*, pages 1–69. Springer-Verlag, 1979.
3. H. Ehrig, K. Ehrig, A. Habel, and K.-H. Pennemann. Theory of constraints and application conditions: From graphs to high-level structures. *Fundamenta Informaticae*, 72, 2006.
4. H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs of Theoretical Computer Science. Springer-Verlag, Berlin, 2006.
5. H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. Parallelism and concurrency in high level replacement systems. *Mathematical Structures in Computer Science*, 1:361–404, 1991.
6. H. Ehrig, A. Habel, J. Padberg, and U. Prange. Adhesive high-level replacement systems: A new categorical framework for graph transformation. *Fundamenta Informaticae*, 72, 2006.
7. A. Habel, J. Müller, and D. Plump. Double-pushout graph transformation revisited. *Mathematical Structures in Computer Science*, 11(5):637–688, 2001.
8. A. Habel and K.-H. Pennemann. Nested constraints and application conditions for high-level structures. In *Formal Methods in Software and System Modeling*, volume 3393 of *Lecture Notes in Computer Science*, pages 293–308. Springer-Verlag, 2005.
9. A. Habel, K.-H. Pennemann, and A. Rensink. Weakest preconditions for high-level programs. In *Proc. Int. Conference on Graph Transformation (ICGT 2006)*, this volume of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.
10. R. Heckel and A. Wagner. Ensuring consistency of conditional graph grammars — a constructive approach. In *SEGRAGRA '95*, volume 2 of *Electronic Notes in Theoretical Computer Science*, pages 95–104, 1995.
11. M. Koch, L. V. Mancini, and F. Parisi-Presicce. Graph-based specification of access control policies. *Journal of Computer and System Sciences*, 71:1–33, 2005.
12. S. Lack and P. Sobociński. Adhesive categories. In *Proc. of Foundations of Software Science and Computation Structures (FOSSACS'04)*, volume 2987 of *Lecture Notes in Computer Science*, pages 273–288. Springer-Verlag, 2004.
13. A. Rensink. Representing first-order logic by graphs. In *Graph Transformations (ICGT'04)*, volume 3256 of *Lecture Notes in Computer Science*, pages 319–335. Springer-Verlag, 2004.