Electronic Communications of the EASST
Volume X (2010)

International Colloquium on Graph and Model Transformation – On the occasion of the 65th birthday of Hartmut Ehrig (GMT 2010)

Expressiveness of graph conditions with variables

Annegret Habel and Hendrik Radke

20 pages

Guest Editors: Claudia Ermel, Hartmut Ehrig, Fernando Orejas, Gabriele Taentzer
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Expressiveness of graph conditions with variables

Annegret Habel[1] **and Hendrik Radke**[2*]

[1] habel@informatik.uni-oldenburg.de
[2] radke@informatik.uni-oldenburg.de
Carl v. Ossietzky Universität Oldenburg, Germany

Abstract: Graph conditions are most important for graph transformation systems and graph programs in a large variety of application areas. Nevertheless, non-local graph properties like "there exists a path", "the graph is connected", and "the graph is cycle-free" are not expressible by finite graph conditions. In this paper, we generalize the notion of finite graph conditions, expressively equivalent to first-order formulas on graphs, to finite HR graph conditions, i.e., finite graph conditions with variables where the variables are place holders for graphs generated by a hyperedge replacement system. We show that graphs with variables and replacement morphisms form a weak adhesive HLR category. We investigate the expressive power of HR graph conditions and show that finite HR graph conditions are more expressive than monadic second-order graph formulas.

Keywords: Graph conditions, graphs with variables, hyperedge replacement systems, monadic-second order graph formulas, weak adhesive HLR categories.

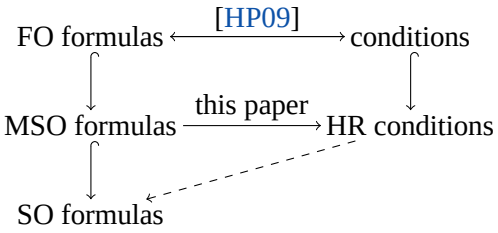
1 Introduction

Graph transformation systems have been studied extensively and applied to several areas of computer science [Roz97, EEKR99, EKMR99] and were generalized to high-level replacement (HLR) systems [EHKP91] and weak adhesive HLR systems [EEPT06b]. Graph conditions, i.e., graph constraints and application conditions, studied e.g. in [EH86, HHT96, HW95, KMP05, EEHP06, HP09], are most important for graph transformation systems and graph programs in a large variety of application areas. Graph conditions are an intuitive, graphical, yet precise formalism, well-suited for describing structural properties. Moreover, finite graph conditions and first-order graph formulas are expressively equivalent [HP09]. Unfortunately, typical graph properties like "there exists a path", "the graph is connected", and "the graph is cycle-free" are not expressible by first-order graph formulas [Cou90, Cou97b] and *finite* graph conditions. They only can be expressed by *infinite* graph conditions.

In this paper, we generalize the concept of graph conditions [HP09] to HR graph conditions, i.e. graph conditions with variables where the variables are place holders for graphs generated by a hyperedge replacement (HR) system. By the HR system, we obtain a finite description of a, in general, infinite set of graphs, e.g., the set of all paths. We investigate the expressive power of HR graph conditions and show that monadic second-order (MSO) graph formulas can be expressed by equivalent HR graph conditions, but also some second-order (SO) graph properties

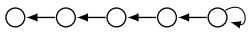
* This work is supported by the German Research Foundation (DFG) under grant GRK 1076/1 (Graduate School on Trustworthy Software Systems).

can be expressed by HR graph conditions.

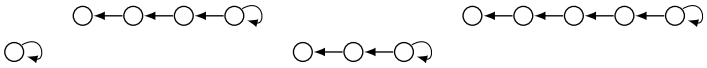


The usefulness of HR conditions is illustrated by an example of a car platooning maneuver protocol.

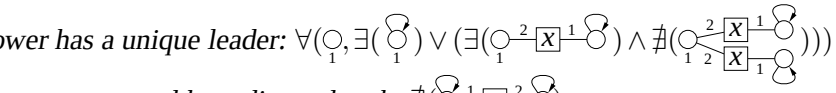
Example 1 (Car platooning) *In the following, we study "a prototypical instance of a dynamic communication system", originally taken from the California Path project [HESV91]. It represents a protocol for cars on a highway that can organize themselves into platoons, by driving close together, with the aim to conserve space and fuel. A car platoon is modeled as a directed graph where the nodes represent the cars and the direct edges the direct following relation. Additionally, the leader of a car platoon is marked by a loop.*

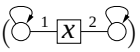


A car platooning state graph consists of zero or more car platoons.



Car platooning operations like splitting a car platoon in two car platoons, or joining two car platoons into a single one can be described by graph replacement rules. When performing these operations, certain car platooning properties have to be satisfied:

(1) Every follower has a unique leader: 

(2) Leaders are not connected by a directed path: 

(3) The car platooning state graph is circle-free: 

with $x ::=$  .

The car platooning properties are described by HR graph conditions. Bauer [Bau06] and Pennemann [Pen09] model the following relation with respect to the leader, but not the direct following relation. HR graph conditions allow to express path conditions as in the car platooning example.

The paper is organized as follows: In Section 2, we introduce graphs with variables. In Section 3, we generalize graph conditions to HR graph conditions, i.e. graph conditions with variables equipped with a hyperedge replacement (HR) system. In Section 4, we present a number of examples for HR conditions. In Section 5, we investigate the expressive power of HR conditions. A conclusion including further work is given in Section 6.

2 Graphs with variables

Graphs with variables consist of nodes, edges, and hyperedges. Edges have one source and one target and are labeled by a symbol of an alphabet; hyperedges have an arbitrary sequence of attachment nodes and are labeled by variables.

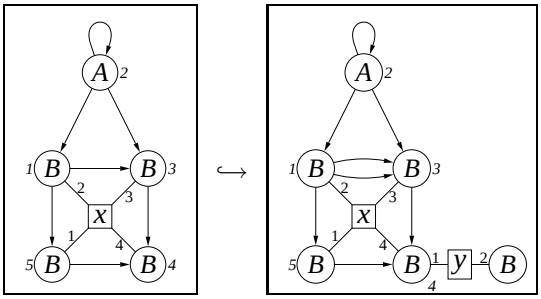
Definition 1 (Graphs with variables) Let $\mathrm{C} = \langle \mathrm{C_V}, \mathrm{C_E}, \mathrm{Var} \rangle$ be a fixed, finite label alphabet where Var is a set of variables with a mapping $\mathrm{rank}\colon \mathrm{Var} \to \mathbb{N}_0$ defining the rank of each variable. A *graph with variables* over C is a system $G = (\mathrm{V}_G, \mathrm{E}_G, \mathrm{Y}_G, \mathrm{s}_G, \mathrm{t}_G, \mathrm{att}_G, \mathrm{lv}_G, \mathrm{le}_G, \mathrm{ly}_G)$ consisting of finite sets V_G, E_G, and Y_G of *nodes* (or *vertices*), *edges*, and *hyperedges*, *source* and *target functions* $\mathrm{s}_G, \mathrm{t}_G\colon \mathrm{E}_G \to \mathrm{V}_G$, an *attachment function* $\mathrm{att}\colon \mathrm{Y}_G \to \mathrm{V}_G^*$, and *labeling functions* $\mathrm{lv}_G\colon \mathrm{V}_G \to \mathrm{C_V}$, $\mathrm{le}_G\colon \mathrm{E}_G \to \mathrm{C_E}$, $\mathrm{ly}\colon \mathrm{Y}_G \to \mathrm{Var}$ such that, for all $y \in \mathrm{Y}_G$, $|\mathrm{att}(y)| = \mathrm{rank}(\mathrm{ly}_G(y))$. For $y \in \mathrm{Y}_G$, $\mathrm{rank}(y) = |\mathrm{att}(y)|$ denotes the *rank* of y. For $\mathrm{Y}_G = \emptyset$, G is a *graph*. The *size* of a graph G is the number of nodes and edges, i.e., $\mathrm{size}(G) = |\mathrm{V}_G| + |\mathrm{E}_G|$. $\mathscr{G}_{\mathrm{Var}}$ denotes the set of all graphs with variables, $\mathscr{G}$ the set of all graphs, and $\mathscr{G}^n$ the set of all graphs of size $\leq n$. For $x \in \mathrm{Var}$ with $\mathrm{rank}(x) = n$, $x^\bullet$ denotes the graph with the nodes $v_1, \ldots, v_n$ and one hyperedge attached to $v_1 \ldots v_n$. A *pointed* graph with variables $\langle R, \mathrm{pin}_R \rangle$ is a graph with variables R together with a sequence $\mathrm{pin}_R = v_1 \ldots v_n$ of pairwise distinct nodes from R. We write $\mathrm{rank}(R)$ for the number n of nodes and Pin_R for the set $\{v_1, \ldots, v_n\}$.

Remark 1 *The definition extends the well-known definition of graphs [Ehr79] by the concept of hyperedges in the sense of [Hab92]. Graphs with variables also may be seen as special hypergraphs where the set of hyperedges is divided into a set of edges labelled with terminal symbols (of* $\mathrm{C_E}$*) and a set of hyperedges labelled by nonterminal symbols (of* Var*).*

We extend the definition of graph morphisms to the case of graphs with variables.

Definition 2 (Graph morphisms with variables) A *(graph) morphism (with variables)* $g\colon G \to H$ consists of functions $g_{\mathrm{V}}\colon \mathrm{V}_G \to \mathrm{V}_H$, $g_{\mathrm{E}}\colon \mathrm{E}_G \to \mathrm{E}_H$, and an injective function $g_{\mathrm{Y}}\colon \mathrm{Y}_G \to \mathrm{Y}_H$ that preserve sources, targets, attachment nodes, and labels, that is, $\mathrm{s}_H \circ g_{\mathrm{E}} = g_{\mathrm{V}} \circ \mathrm{s}_G$, $\mathrm{t}_H \circ g_{\mathrm{E}} = g_{\mathrm{V}} \circ \mathrm{t}_G$, $\mathrm{att}_H = \mathrm{att}_G$, $\mathrm{lv}_H \circ g_{\mathrm{V}} = \mathrm{lv}_G$, $\mathrm{le}_H \circ g_{\mathrm{E}} = \mathrm{le}_G$, and $\mathrm{ly}_H \circ g_{\mathrm{Y}} = \mathrm{ly}_G$. A morphism g is *injective* (*surjective*) if g_{V}, g_{E}, and g_{Y} are injective (surjective), and an *isomorphism* if it is both injective and surjective. In the latter case G and H are *isomorphic*, which is denoted by $G \cong H$. The *composition* $h \circ g$ of g with a graph morphism $h\colon H \to M$ consists of the composed functions $h_{\mathrm{V}} \circ g_{\mathrm{V}}$, $h_{\mathrm{E}} \circ g_{\mathrm{E}}$, and $h_{\mathrm{Y}} \circ g_{\mathrm{Y}}$. For a graph G, the *identity* $\mathrm{id}_G\colon G \to G$ consists of the identities $\mathrm{id}_{G\mathrm{V}}$, $\mathrm{id}_{G\mathrm{E}}$, and $\mathrm{id}_{G\mathrm{Y}}$ on G_{V}, G_{E}, and G_{Y}, respectively.

Example 2 *Consider the graphs G and H over the label alphabet* $\mathrm{C} = \langle \{A, B\}, \{\square\}, \mathrm{Var} \rangle$ *where the symbol* $\square$ *stands for the invisible edge label and is not drawn and* $\mathrm{Var} = \{x, y\}$ *is the set of variables of rank* 4 *and* 2*, respectively. The graph G contains five nodes with the labels A and B, respectively, seven edges with label* $\square$ *which is not drawn, and one hyperedge of rank 4 with label x. Additionally, the graph H contains a node, an edge, and a hyperedge of rank 2 with label y.*

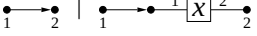


G H

The drawing of graphs with variables combines the drawing of graphs in [Ehr79] and the drawing of hyperedges in [Hab92, DHK97]: Nodes are drawn by circles carrying the node label inside, edges are drawn by arrows pointing from the source to the target node and the edge label is placed next to the arrow, and hyperedges are drawn as boxes with attachment nodes where the i-th tentacle has its number i written next to it and is attached to the i-th attachment node and the label of the hyperedge is inscribed in the box. Arbitrary graph morphisms are drawn by usual arrows "→"; the use of "↪" indicates an injective graph morphism. The actual mapping of elements is conveyed by indices, if necessary.

In [PH96, Pra04], variables are substituted by arbitrary graphs. In this paper, variables are replaced by graphs generated by a hyperedge replacement system. It can be shown that satisfiability by substitution and satisfiability by replacement are expressively equivalent. By our opinion, the second satisfiability notion is the more adequate.

Definition 3 (HR systems) A *hyperedge replacement (HR) system* $\mathscr{R}$ is a finite set of replacement pairs of the form x/R where x is a variable and R a pointed graph with $\mathrm{rank}(x) = \mathrm{rank}(R)$. Given a graph G, the application of the replacement pair x/R to a hyperedge y with label x and $\mathrm{rank}(y) = \mathrm{rank}(x)$ proceeds in two steps: 1. Remove the hyperedge y from G, yielding the graph $G-\{y\}$. 2. Construct the disjoint union $(G-\{y\})+R$ and fuse the i^{th} node in $\mathrm{att}_G(y)$ with the i^{th} attachment point of R, for $i = 1,\ldots,\mathrm{rank}(y)$, yielding the graph H. Then G *directly derives* H by x/R applied to y, denoted by $G \Rightarrow_{x/R,y} H$ or $G \Rightarrow_{\mathscr{R}} H$ provided $x/R \in \mathscr{R}$. A sequence of direct derivations $G \Rightarrow_{\mathscr{R}} \ldots \Rightarrow_{\mathscr{R}} H$ is called a *derivation* from G to H, denoted by $G \Rightarrow^*_{\mathscr{R}} H$. For every variable x, $\mathscr{R}(x) = \{G \in \mathscr{G} \mid x^\bullet \Rightarrow^*_{\mathscr{R}} G\}$ denotes the set of all graphs derivable from $x^\bullet$ by $\mathscr{R}$.

Example 3 *The hyperedge replacement system* $\mathscr{R}$ *with the rules given in Backus-Naur form* $x ::=$  *generates the set of all directed paths from node 1 to node 2.*

Assumption 1 *In the following, let* $\mathscr{R}$ *be a fixed HR-system.*

Hyperedge replacement systems define replacements. A replacement morphism consists of a replacement and a graph morphism.

Definition 4 (Replacement morphisms) A *replacement* is a finite set $\rho = \{y_1/R_1, \ldots, y_n/R_n\}$ of pairs y_i/R_i where, for $i = 1,\ldots,n$, y_i is a hyperedge, $R_i \in \mathscr{R}(\mathrm{ly}(y_i))$ is a pointed graph with $\mathrm{rank}(y_i) = \mathrm{rank}(R_i)$, and $y_1,\ldots,y_n$ are pairwise distinct. The application of ρ to a graph with

variables G yields the graph $\rho(G)$ obtained from G by applying $\mathrm{ly}(y_i)/R_i \in \mathscr{R}$ to y_i for $i = 1,\ldots n$. $\emptyset$ denotes the *empty* replacement. A *(replacement) morphism* $\hat{g} = \langle g, repl\rangle \colon G \to H$ consists of a graph morphism $g \colon G \to H'$ and a replacement with $\rho(H') = H$. It is *injective* if g is injective. $\langle g, \emptyset\rangle$ is a *graph morphism* and $\langle \mathrm{id}_G, \emptyset\rangle$ the *identity*.

Remark 2 *For every replacement morphism $\langle g, \rho\rangle \colon G \to H$ with graph morphism $g \colon G \to H'$ and $\rho(H') = H$, there is a pair $\langle \rho', g'\rangle \colon G \to H$ with replacement $\rho' = \{y/R \mid g(y)/R \in \rho\}$ and a graph morphism with $g'|_{G-\mathrm{Y}_G} = g$ and $g'|_{\rho'(y)} = \mathrm{id}$ for all $y \in \mathrm{Y}_G$. In the following, we often use replacement morphisms consisting of a replacement and a morphism.*

$$\begin{array}{ccc} G & \xrightarrow{g} & H' \\ {\scriptstyle\rho'}\Downarrow & & \Downarrow{\scriptstyle\rho} \\ G' & \xrightarrow[g']{} & H \end{array}$$

Remark 3 *For replacement morphisms $\langle g_1, \rho_1\rangle \colon G \to H$ and $\langle g_2, \rho_2\rangle \colon H \to I$, the* composition *is defined by $\langle g_2' \circ g_1, \rho_2 \circ \rho_1^*\rangle \colon G \to I$ where the graph morphisms and replacements are as in the figure below.*

$$\begin{array}{ccccc} G & \xrightarrow{g_1} & H' & \xrightarrow{g_2'} & I'' \\ & & {\scriptstyle\rho_1}\Downarrow & & \Downarrow{\scriptstyle\rho_1^*} \\ & & H & \xrightarrow{g_2} & I' \\ & & & & \Downarrow{\scriptstyle\rho_2} \\ & & & & I \end{array}$$

Fact 1 *The composition of replacement morphisms is a replacement morphism.*

Notation 1 *Replacements ρ with $\rho(G) = H$ are denoted by $\rho \colon G \Rightarrow H$.*

In [Pra04], Ulrike Prange sketches that graphs and graph morphisms with variables based on substitution form a category and that the category with the class $\mathscr{M}$ of all injective graph morphisms is an adhesive HLR category. Similarly, one can show that graphs with variables and replacement morphisms form a category and the category with the class $\mathscr{M}$ of all injective graph morphisms is a weak adhesive HLR category.

Fact 2 (Category XGraphs) *Graphs with variables and replacement morphisms form the category* XGraphs.

Proof. Consequence of the associativity and identity of replacements and graph morphisms. □

Fact 3 (XGraphs is weak adhesive HLR) *The category $\langle$XGraphs$, \mathscr{M}\rangle$ of graphs with variables with the class $\mathscr{M}$ of all injective graph morphisms is a weak adhesive HLR category.*

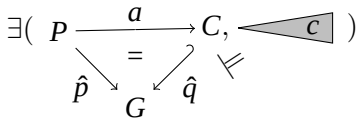
Proof. The proof is given in the Appendix B. □

3 HR conditions

Graph conditions are a well-known concept for describing graph properties in a graphical way by graphs and graph morphisms (see e.g. [EH86, HHT96, HW95, KMP05, EEHP06, HP09]). In the following, we generalize the concept to HR conditions. HR conditions are conditions in the category of graphs with variables where the variables may be replaced by graphs generated by a hyperedge replacement (HR) system. Additionally, HR conditions may contain conditions of the form $P \sqsubseteq C$ where P and C are graphs with variables with the meaning "is subgraph of". This is a counterpart to the MSO subformula $x \in X$ with the meaning "is element in".

Assumption 2 *In the following, let $\mathscr{M}'$ be the class of all injective replacement morphisms.*

Definition 5 (HR conditions) A *condition (with variables)* over a P is of the form true, $P' \sqsubseteq C$ or $\exists(a,c)$, where P', P, and C are graphs with variables, $P' \subseteq P$, $a\colon P \to C$ a morphism, and c a condition over C. Moreover, Boolean formulas over conditions over P are conditions over P. $\exists a$ abbreviates $\exists(a,\text{true})$, $\forall(a,c)$ abbreviates $\neg\exists(a,\neg c)$. A *HR condition* $\langle c,\mathscr{R}\rangle$ consists of a condition with variables c and a HR system $\mathscr{R}$. If $\mathscr{R}$ is clear from the context, we only write the condition c. A HR condition is *finite*, if every conjunction and every disjunction is finite.



Every replacement morphism *satisfies* true. A replacement morphism $\hat{p}\colon P \to G$ *satisfies* $P' \sqsubseteq C$ if $\hat{p}(P') \subseteq \hat{q}(C)$, and $\exists(a,c)$ if there exists a replacement morphism $\hat{q}$ in $\mathscr{M}'$ such that $\hat{q} \circ a = \hat{p}$ and, if c is a condition over C, $\hat{q}$ satisfies c. The satisfaction of conditions by replacement morphisms is extended to Boolean formulas over conditions in the usual way. Every graph *satisfies* true and a graph G *satisfies* the condition c, if c is a condition over $\emptyset$ and the morphism $\emptyset \to G$ satisfies c. We write $\hat{p} \models c$ [$G \models c$] to denote that all replacement morphisms $\hat{p}$ [graphs G] satisfy c. Two conditions c and c' are *equivalent*, denoted by $c \equiv c'$, if, for all replacement morphisms $\hat{p}$, $\hat{p} \models c$ iff $\hat{p} \models c'$.

Remark 4 *The definition generalizes the definitions of conditions in [HHT96, HW95, KMP05, EEHP06, HP09]. In the context of graphs, conditions are also called* constraints *and, in the context of rules, conditions are also called* application conditions. *The generalization is twofold:*

(1) Variables in HR conditions are allowed. The variables are replaced by graphs generated by a corresponding HR system. By this generalization, several infinite *conditions can be expressed by a* finite *HR conditions.*

(2) Conditions of the form $P \sqsubseteq C$ are allowed. Typical examples are are $\overset{x}{\bullet} \sqsubseteq \boxed{X}$ with $X ::= \emptyset \mid \boxed{X}\bullet$ and $\bullet\!\!\rightarrowtail\!\!\bullet \sqsubseteq \boxed{X}$ with $X ::= \emptyset \mid \boxed{X} \bullet\!\!\rightarrowtail\!\!\bullet$. By this generalization, there is a transformation of MSO formulas into equivalent HR conditions.

Example 4 *Consider the HR conditions*

$$\begin{array}{lcl} \texttt{path}(1,2) & = & \exists(\bullet_1\ \bullet_2 \to \bullet_1 \overset{1}{\boxed{X}}\overset{2}{} \bullet_2) \\ \texttt{connected} & = & \forall(\emptyset \to \bullet_1\ \bullet_2, \texttt{path}(1,2)) \\ \texttt{cyclefree} & = & \nexists(\emptyset \to \bullet_1 \overset{1}{\underset{2}{\boxed{X}}}) \\ \texttt{hamiltonian} & = & \exists(\emptyset \to \bullet_1 \overset{1}{\underset{2}{\boxed{X}}}, \nexists(\bullet_1 \overset{1}{\underset{2}{\boxed{X}}} \to \bullet_1 \overset{1}{\underset{2}{\boxed{X}}} \bullet)) \end{array}$$

with the HR system $x ::= \bullet_1 \to \bullet_2 \mid \bullet_1 \to \bullet_1 \overset{1}{\boxed{X}}\overset{2}{} \bullet_2$. *A morphism with domain* $\bullet_1\ \bullet_2$ *satisfies the condition* $\texttt{path}(1,2)$ *iff there exists a path from the image of* 1 *to the image of* 2 *in the range. A graph satisfies* `connected` *iff, for each pair of distinct nodes, there is a nonempty path, i.e., the graph is strongly connected. It satisfies* `cyclefree` *iff there does not exist a cycle, i.e.,the graph is cycle-free. A graph satisfies* `hamiltonian` *iff there exists a circuit and there is no additional node outside the circuit, i.e., the graph is hamiltonian.*

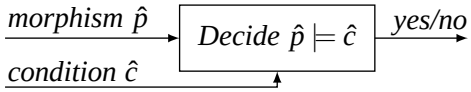
Notation 2 *For a morphism* $a\colon P \to C$ *in a condition, we just depict C, if P can be unambiguously inferred, i.e. for constraints over the empty graph* $\emptyset$ *and for conditions over some left- or right-hand side of a rule. E.g. the constraint* `hamiltonian` *has the short notation* $\exists(\bullet_1 \overset{1}{\underset{2}{\boxed{X}}}, \nexists(\bullet_1 \overset{1}{\underset{2}{\boxed{X}}} \bullet))$.

In the following, we investigate the model checking problem for HR conditions:

Given:	A HR condition c and a graph G
Question:	$G \models c$?

For finite HR conditions, the model checking problem is decidable.

Theorem 1 (Decidability of the model checking problem) *For every finite HR condition* $\hat{c}$ *and every morphism* $\hat{p}$ *[graph G], it is decidable whether or not* $\hat{p} \models \hat{c}$ *[*$G \models \hat{c}$*].*



Proof. Let $\hat{c} = \langle c, \mathscr{R} \rangle$ be a finite HR condition. Without loss of generality, we may assume that $\mathscr{R}$ is monotone, i.e., for each rule $x/R \in \mathscr{R}$, $\text{size}(x^\bullet) \leq \text{size}(R)$. Otherwise, we transform $\mathscr{R}$ into an equivalent monotone HR system (see [Hab92], Theorem 1.5). Let $\hat{p} = \langle \rho_0, p \rangle \colon P \to G$ be a morphism. Suppose that $\text{size}(G) = n$ for some $n \geq 0$. Let Repl denote the set of all replacements. For all $x \in \text{Var}$, $n \in \mathbb{N}_0$, and $C \in \mathscr{G}_{\text{Var}}$, the sets

$$\begin{array}{lcl} \mathscr{R}^n(x) & = & \{G \in \mathscr{G}^n \mid x^\bullet \Rightarrow^*_{\mathscr{R}} G\} \\ \text{Repl}^n(C) & = & \{\rho \in \text{Repl} \mid \rho(y) \in \mathscr{R}^n(\text{ly}(y)) \text{ for all } y \in \text{Y}_C\} \\ \mathscr{R}^n(C) & = & \{\rho \in \text{Repl}^n(C) \mid \rho(C) \leq n\} \end{array}$$

can be constructed effectively. For $x \in \text{Var}$ and $k \in \mathbb{N}$, define sets $\mathscr{R}^n_k(x)$ recursively as follows: $\mathscr{R}^n_1(x) = \{R \in \mathscr{G}_n \mid x/R \in \mathscr{R}\}$ and, for $k \geq 1$, $\mathscr{R}^n_{k+1}(x) = \mathscr{R}^n_k(x) \cup \{\rho(R) \in \mathscr{G}^n \mid x/R \in \mathscr{R}, \rho \in \text{Repl}^n_k(R)\}$ where $\text{Repl}^n_k(R) = \{\rho \in \text{Repl} \mid \rho(y) \in \mathscr{R}^n_k(\text{ly}(y)) \text{ for all } y \in \text{Y}_R\}$. Since $\mathscr{R}^n_k(x) \subseteq \mathscr{R}^n_{k+1}(x) \subseteq \mathscr{G}^n$ for all $k \in \mathbb{N}$ and $\mathscr{G}^n$ is finite, there is some $l(x) \in \mathbb{N}$ such that $\mathscr{R}^n_{l(x)}(x) =$

$\mathscr{R}^n_{l(x)+1}(x)$. This implies $\mathscr{R}^n_{l(x)}(x) = \mathscr{R}^n_{l(x)+m}(x)$ for all $m \geq 1$. Now all $\mathscr{R}^n_k(x)$ up to the smallest possible $l(x)$ can be constructed effectively. Furthermore, $\mathscr{R}^n(x) = \mathscr{R}^n_{l(x)}(x)$ may be verified. $\mathscr{R}^n(x) = \bigcup_{k=1}^{\infty} \mathscr{R}^n_k$. Since the sets $\mathscr{R}^n_k(x)$ are monotonically increasing subsets of $\mathscr{G}^n$ and since $\mathscr{G}^n$ is finite, there is some $l(x) \in \mathbb{N}$ such that $\mathscr{R}^n_{l(x)}(x) \subseteq \mathscr{R}^n_{l(x)+1}(x)$. By definition, $\mathscr{R}^n_{l(x)}(x) = \mathscr{R}^n_{l(x)+1}(x) = \mathscr{R}^n_{l(x)+2}(x) = \ldots$. Thus, $\bigcup_{k=1}^{\infty} \mathscr{R}^n_k = \mathscr{R}^n_{l(x)}(x)$. The second and third statement follow directly from the first one.

For "existential" HR conditions $\hat{d} = \langle d, \mathscr{R} \rangle$ with $d = \exists(a,c)$ with $a\colon P \to C$,

$$\boxed{\hat{p} \models \langle d, \mathscr{R} \rangle \iff p \models \bigvee_{\rho \in \mathscr{R}^n(C)} \rho(d).}$$

$$\begin{array}{lll} \hat{p} \models \hat{d} & \iff & \exists \hat{q} = \langle \rho, q \rangle \in \mathscr{M}'.\hat{q} \circ a = \hat{p} \wedge \hat{q} \models c \\ & \iff & \exists \rho \in \mathrm{Repl}.\exists q \in \mathscr{M}.q \circ \rho(a) = p \wedge q \models \rho(c) \\ & \iff & p \models \bigvee_{\rho \in \mathrm{Repl}} \rho(d) \\ & \iff & p \models \bigvee_{\rho \in \mathscr{R}^n(C)} \rho(d). \end{array}$$

Since $\mathscr{R}^n(C)$ is finite, $\bigvee_{\rho \in \mathscr{R}^n(C)} \rho(d)$ is a finite. For all existential subconditions, satisfiability can be tested and, for non-existential conditions, satisfiability can be inferred. For a graph G and a HR condition $\langle d, \mathscr{R} \rangle$ where d is a condition over $\emptyset$, $G \models \langle d, \mathscr{R} \rangle \iff \emptyset \to G \models \langle d, \mathscr{R} \rangle$. □

Remark 5 *Theorem 1 makes use of the monotonicity property of HR systems. Allowing monotone replacement system instead of (monotone) HR systems one would get a corresponding decidability result.*

4 A classification of graph properties

By a *graph property*, we mean a predicate on the class of graphs that is stable under isomorphism. In the following, we collect a number of graph properties known to be first order, monadic second-order, and second order, respectively, and show that most of them can be expressed by HR conditions.

Fact 4 (classification of graph properties [Cou90]) *The following properties of a directed, labelled graph G and nodes v and w are first order (FO), monadic second-order (MSO), and second order (SO), respectively:*

properties of a directed, labelled graph G	*FO*	*MSO*	*SO*
– simple	*yes*	*yes*	*yes*
– k-regular	*yes*	*yes*	*yes*
– degree $\leq k$	*yes*	*yes*	*yes*
– has a nonempty path from v to w	*no*	*yes*	*yes*
– (strongly) connected	*no*	*yes*	*yes*
– planar	*no*	*yes*	*yes*
– k-colorable	*no*	*yes*	*yes*
– Hamiltonian	*no*	*yes*	*yes*
– is a tree	*no*	*yes*	*yes*
– is a square grid	*no*	*yes*	*yes*
– has an even number of nodes	*no*	*no*	*yes*
– has as many edges labelled a as b	*no*	*no*	*yes*
– has a nontrivial automorphism	*no*	*no*	*yes*
– $\mathrm{card}(V_G)$ *belongs to a given nonrecursive set*	*no*	*no*	*no*

The following monadic second-order graph properties can be expressed by HR conditions.

Example 5 (MSO graph properties) *For the hyperedge replacement system with the rules* $x ::= \bullet_1 \to \bullet_2 \mid \bullet_1 \to \bullet \xrightarrow{1}\boxed{X}\xrightarrow{2} \bullet_2$ *, we have the following:*

1. **Paths.** *A* nonempty path *in G is here a sequence of nodes* $(v_1, v_2, \ldots, v_n)$ *with* $n \geq 2$ *such that there is an edge with source* v_i *and target* v_{i+1} *for all i and* $v_i = v_j \Rightarrow \{i, j\} = \{1, n\}$*; if* $v_1 = v_n$*, this path is a* circuit. *The HR condition* $\mathtt{path}(1,2) = \exists(\ {}_1\bullet\ {}_2\bullet \to \bullet_1 \xrightarrow{1}\boxed{X}\xrightarrow{2}\bullet_2\ )$ *requires that there is a nonempty path from the image of 1 to the image of 2.*

2. **Connectedness.** *The HR condition* $\mathtt{connected} = \forall({}_1\bullet\ {}_2\bullet, \mathtt{path}(1,2))$ *requires that, for each pair of distinct nodes, there is a nonempty path, i.e., the graph is strongly connected.*

3. **Cycle-freeness.** *The HR condition* $\mathtt{cyclefree} = \nexists(\bullet_1 \xleftarrow{1}\boxed{X}\xrightarrow{2}\bullet_2)$ *requires that the graph is cycle-free.*

4. **Planarity.** *By Kuratowski's Theorem (see e.g. [Eve79]) a graph is* planar *if and only if it has no subgraph homeomorphic to* $K_{3,3}$ *or* K_5*. Two graphs are* homeomorphic *if both can be obtained from the same graph by insertion of new nodes of degree 2, in edges, i.e. an edge is replaced by a path whose intermediate nodes are all new. The HR condition* $\mathtt{planar} = \nexists(K_5^*) \wedge \nexists(K_{3,3}^*)$ *where* K_5^* *and* $K_{3,3}^*$ *are obtained from the graphs* K_5 *and* $K_{3,3}$ *by replacing all edges by hyperedges with label x, respectively, requires that the graph has no subgraph homeomorphic to* K_5 *or* $K_{3,3}$*, i.e. that the graph is planar.*

5. **Coloring.** *A* coloring *of a graph is an assignment of colors to its nodes so that two adjacent nodes have the same color. A k-coloring of a graph G uses k colors. By König's characterization (see e.g. [Har69]), a graph is 2-colorable if and only if it has no odd cycles. For undirected graphs, i.e., graphs in which each undirected edge stands for two directed edges in opposite direction, the HR condition* $2\mathtt{color} = \nexists(\bullet_1 \xleftrightarrow{1}\boxed{X}\xrightarrow{2}\bullet_2)$ *with* $x ::= \bullet_1 \to \bullet_2 \mid \bullet_1 \to \bullet \xrightarrow{1}\boxed{X}\xrightarrow{2}\bullet_2$ *requires that there are no cycles of odd length, i.e., the graph is 2-colorable.*

6. **Hamiltonicity.** *A graph is* Hamiltonian*, if there exists a Hamiltonian circuit, i.e. a simple circuit on which every node of the graph appears exactly once (see e.g. [Eve79]). For undirected graphs, the HR condition* $\texttt{hamiltonian} = \exists(\ \boxed{x}\ , \nexists(\ \boxed{x}\ \bullet))$ *with* $x ::= \ \bullet\!-\!\bullet \ \mid\ \bullet\!-\!\bullet\!-\!\boxed{x}\!-\!\bullet$ *requires that there exists a simple circuit in the graph and there is no additional node in the graph, i.e. every node of the graph lies on the circuit, the graph is Hamiltonian.*

7. **Trees.** *A graph G is a* tree *if is connected, cycle-free, and has a root. A graph G has a* root *v if v is a node in G and every other node v′ in G is* reachable *from v, i.e. there is a directed path from v to v′ (see e.g. [Eve79]). For undirected graphs, the HR condition* $\texttt{tree} = \texttt{uconnected} \wedge \texttt{ucyclefree} \wedge \exists(\bullet, \forall(\bullet\ \bullet, \texttt{upath}(1,2)))$ *(with the undirected versions of* `connected`*,* `cyclefree`*, and* $\texttt{path}(1,2)$*) requires that the graph is connected, cycle-free, and has a root, i.e., the graph is a tree.*

The following second-order graph properties can be expressed by HR graph conditions.

Example 6 (SO graph properties)

1. **Even number of nodes.** *The HR condition* $\texttt{even} = \exists(\ \boxed{x}\ , \nexists(\ \boxed{x}\ \bullet))$ *with* $x ::= \ \emptyset \mid \bullet\ \bullet\ \boxed{x}$ *expresses the SO graph property "the graph has an even number of nodes".*

2. **Equal number of** *a***'s and** *b***'s.** *The HR condition* $\texttt{equal} = \exists(\ \boxed{x}\ , \nexists(\ \boxed{x}\ \textcircled{a}\ ) \wedge \nexists(\ \boxed{x}\ \textcircled{b}\ ))$ *with* $x ::= \ \emptyset \mid \textcircled{a}\ \textcircled{b}\ \boxed{x}$ *expresses the SO graph property "the graph has as many nodes labelled a as b".*

3. **Paths of same length.** *The HR condition* $\texttt{2paths}_{1,2} = \exists(\ \bullet\ \bullet \rightarrow \bullet\!-\!\boxed{x}\!-\!\bullet\ )$ *with* $x ::= \ 1\!\bullet\!\rightarrow\!\bullet 2 \ \atop 3\!\bullet\!\rightarrow\!\bullet 4 \ \mid\ 1\!\bullet\!\rightarrow\!\bullet\!-\!\boxed{x}\!-\!\bullet\!\rightarrow\!\bullet 2 \ \atop 3\!\bullet\!\rightarrow\!\bullet\!\rightarrow\!\bullet 4$ *expresses the SO graph property "there exist two node-disjoint paths of same length from the image of 1 to the image of 2".*

5 Expressiveness of HR conditions

We are interested in classifying HR conditions, i.e. we want to classify the kind of graph properties that can be expressed by HR conditions. To this effect, we compare HR conditions and monadic second-order formulas on graphs [Cou90, Cou97a]. We show that there is transformations from MSO formulas into equivalent HR conditions. Vice versa, there is no such a transformation: HR graph conditions can express second-order graph properties.

Let Rel be a finite set of relation symbols. Let Var contain individual variables and relation variables of arity one. Since a relation with one argument is nothing but a set, we call these variables *set variables*. A *monadic second-order formula* over Rel is a second-order formula written with Rel and Var: the quantified and free variables are individual or set variables; there is no restriction on the arity of symbols in Rel. In order to get more readable formulas, we shall write $x \in X$ instead of $X(x)$ where X is a set variable.

Definition 6 (MSO graph formulas) Let Var be a countable set of individual and set variables. The set of all *monadic second-order (MSO) graph formulas* (over Var) is inductively defined:

For $b \in C$ and $x,y,z,X \in Var$, $l_b(x)$, $inc(x,y,z)$, $x = y$, and $x \in X$ are formulas. For formulas F, F_i ($i \in I$) and variables $x,X \in Var$, $\neg F$, $\wedge_{i\in I} F_i$, $\exists xF$, and $\exists XF$ are formulas. For a formula F, $Free(F)$ denotes the set of all *free* variables of F. A formula is *closed*, if $Free(F) \neq \emptyset$ does not contain free variables. The *semantic* $G[\![F]\!](\sigma)$ of a formula F in a non-empty graph G under assignment $\sigma\colon Var \to V_G + E_G$ is inductively defined as follows. The semantics of the formulas $l_b(x)$, $inc(x,y,z)$, $x = y$, $\neg F$, $\wedge_{i\in I} F_i$, and $\exists xF$ in G under σ is defined as usual (see [HP09]). $G[\![x \in X]\!](\sigma) = true$ iff $\sigma(x) \in \sigma(X)$ and $G[\![\exists XF]\!](\sigma) = true$ iff $G[\![F]\!](\sigma\{X/D\}) = true$ for some $D \subseteq V_G$ or $D \subseteq E_G$ where $\sigma\{X/D\}$ is the modified assignment with $\sigma\{X/D\}(X) = D$ and $\sigma\{X/D\}(x) = \sigma(x)$ otherwise. A graph G *satisfies* a formula F, denoted by $G \models F$, iff for all assignments $\sigma\colon Var \to D_G$, $G[\![F]\!](\sigma) = true$.

Notation 3 *The expression $\vee_{i\in I} F_i$ abbreviates the formula $\neg \wedge_{i\in I} \neg F_i$, $F \Rightarrow G$ abbreviates $\neg F \vee G$, $\forall xF$ abbreviates $\neg\exists x\neg F$, and $\forall XF$ abbreviates $\neg\exists X\neg F$.*

Example 7 *The MSO formula $F_0(x_1,x_2) = \forall X[\{\forall y\forall z(y \in X \wedge edg(y,z) \Rightarrow z \in X) \wedge \forall y(edg(x_1,y) \Rightarrow y \in X)\} \Rightarrow x_2 \in X]$ [Cou97a] expresses the property "There is a nonempty path from x_1 to x_2" . The formula $F_1 = (x = y) \vee F_0(x,y)$ expresses the property "$x = y$ or there is a nonempty path from x to y" and the formula $F_2 = \forall x,y[F_1(x,y)]$ expresses the property "the graph is strongly connected".*

MSO graph formulas can be transferred in equivalent finite HR graph conditions.

Theorem 2 (From MSO formulas to HR conditions) *There is a transformation $Cond_{\mathscr{M}}$ from MSO formulas to HR conditions, such that, for all MSO graph formulas F and all graphs G, $G \models F \Leftrightarrow G \models Cond_{\mathscr{M}}(F)$.*

Proof. Let F be a MSO formula. Without loss of generality, we may assume that F is closed and rectified, i.e. distinct quantifiers bind occurrences of distinct variables; otherwise, we build the universal closure of F and rename the variables. Since F is rectified, the variables of F can be represented by isolated nodes, edges, and hyperedges in the graphs of a constructed condition. Let P be a graph with variables. If the set $D'_P = Iso_P + E_P + Y_P]$, the set of all isolated nodes, edges, and hyperedges in P, is a subset of the set Var of variables, then every morphism $\hat{p}\colon P \to G$ into a non-empty graph G induces an assignment $\sigma\colon Var \to D_G$ such that $\hat{p} = \sigma[D'_P]$, i.e., $\hat{p}(x) = \sigma(x)$ for each $x \in D'_P$. Vice versa, an assignment $\sigma\colon Var \to D_G$ induces a mapping $D'_P \to D_G$ that may be extended to a graph morphism $\hat{p}\colon P \to G$ with $\hat{p} = \sigma[D'_P]$.

$$\begin{array}{ccc} & & Free(F) \\ & & \cap| \\ Var & \supseteq & D'_P \cdots P \\ \sigma & = & \downarrow \quad\quad \downarrow \hat{p} \\ D_G & & \cdots\cdots G \end{array}$$

The HR condition is given by $Cond(F) = Cond(\emptyset, F)$, where $\emptyset$ denotes the empty graph. For a MSO formula F and a graph P with $Free(F) \subseteq D'_P \subseteq Var$, the HR condition $Cond(P,F)$ is constructed as follows: For the expressions known from first-order theory, we use the construction

in [HP09]. For the atomic MSO expressions $x \in X$ and $\exists XF$, where x is an individual variable and X a set variable, we define the transformation as follows:

$$\begin{array}{lcl} \mathrm{Cond}(P, x \in X) & = & \begin{cases} \overset{x}{\bullet} \sqsubseteq \boxed{X} & \text{if } x \in \mathrm{V}_P \\ \bullet \xrightarrow{x} \bullet \sqsubseteq \boxed{X} & \text{if } x \in \mathrm{E}_P \end{cases} \\ \mathrm{Cond}(P, \exists XF) & = & \bigvee_i \langle \exists(P \to C, \mathrm{Cond}(C,F)), \mathscr{R}_i \rangle \text{ where} \\ & & C = P + \boxed{X}, \mathscr{R}_1 : X ::= \emptyset \mid \boxed{X} \bullet \text{ and } \mathscr{R}_2 : X ::= \emptyset \mid \boxed{X} \bullet \!\!\to\!\!\bullet \end{array}$$

The following claim is based on $\mathscr{A}'$-*satisfiability* obtained from the usual satisfiability by replacing all occurrences of $\mathscr{M}'$ by $\mathscr{A}'$, the class of all replacement morphisms. We write $\models_{\mathscr{A}}$ to denote $\mathscr{A}'$-satisfiability. For all rectified MSO graph formulas F, all graphs G, all assignments $\sigma\colon \mathrm{Var} \to D_G$, and all morphisms $\hat{p}\colon P \to G$ with $\sigma = \hat{p}[D'_P]$, $G[\![F]\!](\sigma) = \mathit{true} \Leftrightarrow \hat{p} \models_{\mathscr{A}} \mathrm{Cond}(P,F)$. The proof makes use of the proof of the corresponding statement for rectified FO formulas given in [HP09] and is done by structural induction.

Basis. For the atomic formulas $\mathrm{l}_b(x)$, $\mathrm{inc}(x,y,z)$, and $x = y$, the proof is as in [HP09]. For atomic formulas of the form $x \in X$, the statement follows directly from the definitions:

$$\begin{array}{lll} & G[\![x \in X]\!](\sigma) = \mathit{true} & \\ \Leftrightarrow & \sigma(x) \in \sigma(X) & \text{(Semantics of } x \in X) \\ \Leftrightarrow & \hat{p}(\overset{\bullet}{x}) \subseteq \hat{p}(\boxed{X}) & (\hat{p} = \sigma[D'_P],\ x, X \in \mathrm{Free}(x \in X) \in \mathrm{Free}(x \in X) \subseteq D'_P) \\ \Leftrightarrow & \hat{p} \models_{\mathscr{A}} \overset{\bullet}{x} \sqsubseteq \boxed{X} & \text{(Semantics of } \overset{\bullet}{x} \sqsubseteq \boxed{X}) \\ \Leftrightarrow & \hat{p} \models_{\mathscr{A}} \mathrm{Cond}(P, x \in X) & \text{(Definition of Cond)} \end{array}$$

Hypothesis. Assume, the statement holds for rectified formulas F.

Step. For formulas of the form $\neg F$, $\wedge_{i \in I} F_i$, $\exists xF$, the proof is as in [HP09]. For formulas of the form $\exists XF$, graphs G, and assignments σ, the statement follows from the definitions and the induction hypothesis:

$$\begin{array}{lll} & G[\![\exists XF]\!](\sigma) = \mathit{true} & \\ \Leftrightarrow & \exists D \subseteq D_G . G[\![F]\!](\sigma\{X/D\}) = \mathit{true} & \text{(Semantics of } \exists XF) \\ \Leftrightarrow & \exists D \subseteq \mathrm{V}_G . G[\![F]\!](\sigma\{X/D\}) = \mathit{true} \text{ or} & \\ & \exists D \subseteq \mathrm{E}_G . G[\![F]\!](\sigma\{X/D\}) = \mathit{true} & \text{(Assignment)} \\ \Leftrightarrow & \exists \hat{q}\colon C \to G \in \mathscr{A} . \hat{p} = \hat{q} \circ a \wedge \hat{q} \models_{\mathscr{A}} \mathrm{Cond}(C,F) & \text{(Hypothesis, } \hat{q} = \sigma\{X/D\}[D'_P]) \\ \Leftrightarrow & \hat{p} \models_{\mathscr{A}} \exists(P \to C, \mathrm{Cond}(C,F)) & \text{(Definition } \models_{\mathscr{A}},\ \hat{p} = \sigma[D'_P]) \\ \Leftrightarrow & \hat{p} \models_{\mathscr{A}} \mathrm{Cond}(P, \exists XF) & \text{(Definition Cond)} \end{array}$$

where $a\colon P \to C$ with $C = P + \boxed{X}$.

$\mathscr{A}$-satisfiability is closely related to the satisfiability of monadic second-order graph formulas. As in [HP09], there is a transformation from $\mathscr{A}'$- to $\mathscr{M}'$-satisfiability. [from $\mathscr{A}'$- to $\mathscr{M}'$-satisfiability [HP09]] There is a transformation Msat such that, for every condition c over $\emptyset$ and every graph G, $G \models_{\mathscr{A}'} c \Leftrightarrow G \models_{\mathscr{M}'} \mathrm{Msat}(c)$. . Let $\langle \rho, p \rangle$ be a replacement morphism, $\langle p', \rho' \rangle$ the corresponding pair consisting of a morphism and a replacement, and $\hat{c} = \langle c, \mathscr{R} \rangle$ a HR condition. By the corresponding theorem in [HP09], $p' \models_{\mathscr{A}} c \Leftrightarrow p' \models \mathrm{Msat}(c)$. By the definition of the classes of replacement morphisms $\mathscr{A}'$ and $\mathscr{M}'$, $\langle p', \rho' \rangle \models_{\mathscr{A}'} c \Leftrightarrow \langle p', \rho' \rangle \models \mathrm{Msat}(c)$ and, for the corresponging pair $\langle p^*, \rho^* \rangle$, $\langle p^*, \rho^* \rangle \models_{\mathscr{A}'} c \Leftrightarrow \langle p^*, \rho^* \rangle \models \mathrm{Msat}(c)$. As a consequence, for every graph G, $\mathscr{G} \models_{\mathscr{A}'} c \Leftrightarrow G \models \mathrm{Msat}(c)$. Now, for all graphs G and all closed, rectified MSO formulas F, we have:

$$\begin{array}{lcll} G \models F & \Leftrightarrow & \forall \sigma\colon \mathrm{Var} \to D_G.G[\![F]\!](\sigma) = \mathit{true} & \text{(Definition } \models\text{)} \\ & \Leftrightarrow & \emptyset \to G \models_{\mathscr{A}} \mathrm{Cond}(\emptyset, F) & \text{(Claim 5)} \\ & \Leftrightarrow & G \models_{\mathscr{A}} \mathrm{Cond}(F) & \text{(Definition } \models_{\mathscr{A}}\text{, Cond)} \\ & \Leftrightarrow & G \models \mathrm{Msat}(\mathrm{Cond}(F)) & \text{(Claim 5).} \end{array}$$

Now, the HR condition $\mathrm{Cond}_{\mathscr{M}}(F) = \mathrm{Msat}(\mathrm{Cond}(F))$ has the wanted property. □

Example 8 *The closure of the MSO graph formula*

$$F(x_1,x_2) = \forall X[\{\underbrace{\forall y \forall z(y \in X \wedge \mathrm{edg}(y,z) \Rightarrow z \in X)}_{G_1} \wedge \underbrace{\forall y'(\mathrm{edg}(x_1,y') \Rightarrow y' \in X)}_{G_2}\} \Rightarrow \underbrace{x_2 \in X}_{G_3}]$$

is transformed into the HR condition

$$\begin{array}{ll} & \mathrm{Cond}(\forall x_1 \forall x_2 F(x_1,x_2)) \\ = & \mathrm{Cond}(\emptyset, \forall x_1 \forall x_2 F(x_1,x_2)) \\ \equiv & \forall(\overset{\bullet}{x_1}\overset{\bullet}{x_2}, \mathrm{Cond}(\emptyset, \forall X[G_1 \wedge G_2 \Rightarrow G_3])) \\ \equiv & \forall(\overset{\bullet}{x_1}\overset{\bullet}{x_2}\boxed{X}, [\mathrm{Cond}(\boxed{X}, G_1 \wedge G_2 \Rightarrow G_3]) \\ = & \forall(\overset{\bullet}{x_1}\overset{\bullet}{x_2}\boxed{X}, [\mathrm{Cond}(\boxed{X}, G_1) \wedge \mathrm{Cond}(\boxed{X}, G_2) \Rightarrow \mathrm{Cond}(\boxed{X}, G_3)]) \\ \equiv & \forall(\overset{\bullet}{x_1}\overset{\bullet}{x_2}\boxed{X}, [\,\forall(\overset{\bullet}{x_1}\overset{\bullet}{x_2}\boxed{X}\overset{\bullet}{y}\overset{\bullet}{z}, (\overset{\bullet}{y} \sqsubseteq \boxed{X} \wedge \exists(\overset{\bullet}{x_1}\overset{\bullet}{x_2}\boxed{X}\overset{\bullet}{y}\!\!\longrightarrow\!\!\overset{\bullet}{z}) \Rightarrow \overset{\bullet}{z} \sqsubseteq \boxed{X}) \wedge \\ & \forall(\overset{\bullet}{x_1}\overset{\bullet}{x_2}\boxed{X}\overset{\bullet}{y'}, \exists(\overset{\bullet}{x_1}\!\!\longrightarrow\!\!\overset{\bullet}{y'}\overset{\bullet}{x_2}\boxed{X}\overset{\bullet}{y'}) \Rightarrow \overset{\bullet}{y'} \sqsubseteq \boxed{X}) \Rightarrow \overset{\bullet}{x_2} \sqsubseteq \boxed{X}\,]\,) \end{array}$$

with $X ::= \emptyset \mid \boxed{X}\bullet$ *using the equivalence* $\forall(x(\forall(y,c)) \equiv \forall(y \circ x, c))$ *in [HP05].*

Inspecting the proof of Theorem 2, one may see that only rules of the form $X ::= \emptyset \mid \boxed{X}\bullet$ of $X ::= \emptyset \mid \boxed{X}\bullet\!\!\longrightarrow\!\!\bullet$ are used. A HR condition with rules of this form is called *HR0 condition*.

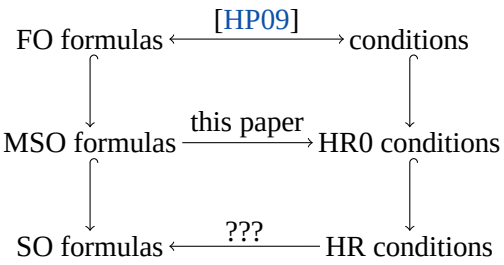
Corollary 1 *There is a transformation* $\mathrm{Cond}_{\mathscr{M}}$ *from MSO formulas to HR0 conditions, such that, for all MSO graph formulas* F *and all graphs* G, $G \models F \Leftrightarrow G \models \mathrm{Cond}_{\mathscr{M}}(F)$.

In Example 6, second-order graph properties are expressed by finite HR conditions. As a consequence, we obtain the following.

Corollary 2 *There is no transformation from finite HR conditions to equivalent MSO formulas.*

6 Conclusion

In this paper, we have generalized the notion of graph conditions to the one of HR graph conditions. The variables in the graphs can to be replaced by graphs generated by a assigned hyperedge replacement system. It is shown that there is a transformation from MSO formulas to HR conditions, but HR conditions are more powerful: they can express certain SO formulas. It remains the question whether or not there are transformations from HR0 conditions to MSO formulas and from HR conditions to SO formulas, respectively.



Graphs with variables and all replacement morphisms form a category. Distinguishing the class of all injective replacement morphisms, we obtain a weak adhesive HLR category. As a consequence, we have

- conditions with variables,
- rules with variables as in [PH96],
- rules with application conditions based on graphs with variables.

We can adapt all results known for weak adhesive HLR categories.

Bibliography

[AHS90] J. Adámek, H. Herrlich, G. Strecker. *Abstract and Concrete Categories*. John Wiley, New York, 1990.

[Bau06] J. Bauer. *Analysis of Communication Topologies by Partner Abstraction*. PhD thesis, Universität Saarbrücken, 2006.

[Cou90] B. Courcelle. Graph Rewriting: An Algebraic and Logical Approach. In *Handbook of Theoretical Computer Science*. Volume B, pp. 193–242. Elsevier, Amsterdam, 1990.

[Cou97a] B. Courcelle. The Expression of Graph Properties and Graph Transformations in Monadic Second- Order Logic. In *Handbook of Graph Grammars and Computing by Graph Transformation*. Volume 1, pp. 313–400. World Scientific, 1997.

[Cou97b] B. Courcelle. On the expression of graph properties in some fragments of monadic second-order logic. In Immerman and Kolaitis (eds.), *Descriptive complexity and finite models*. Pp. 33–62. DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences, 1997.

[DHK97] F. Drewes, A. Habel, H.-J. Kreowski. Hyperedge Replacement Graph Grammars. In *Handbook of Graph Grammars and Computing by Graph Transformation*. Volume 1, pp. 95–162. World Scientific, 1997.

[EEHP06] H. Ehrig, K. Ehrig, A. Habel, K.-H. Pennemann. Theory of Constraints and Application Conditions: From Graphs to High-Level Structures. *Fundamenta Informaticae* 74(1):135–166, 2006.

[EEKR99] H. Ehrig, G. Engels, H.-J. Kreowski, G. Rozenberg (eds.). *Handbook of Graph Grammars and Computing by Graph Transformation*. Volume 2: Applications, Languages and Tools. World Scientific, 1999.

[EEPT06a] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. Fundamental Theory of Typed Attributed Graph Transformation based on Adhesive HLR-Categories. *Fundamenta Informaticae* 74(1):31–61, 2006.

[EEPT06b] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs of Theoretical Computer Science. Springer, Berlin, 2006.

[EH86] H. Ehrig, A. Habel. Graph Grammars with Application Conditions. In Rozenberg and Salomaa (eds.), *The Book of L*. Pp. 87–100. Springer, Berlin, 1986.

[EHKP91] H. Ehrig, A. Habel, H.-J. Kreowski, F. Parisi-Presicce. Parallelism and Concurrency in High Level Replacement Systems. *Mathematical Structures in Computer Science* 1:361–404, 1991.

[Ehr79] H. Ehrig. Introduction to the Algebraic Theory of Graph Grammars. In *Graph-Grammars and Their Application to Computer Science and Biology*. LNCS 73, pp. 1–69. Springer, 1979.

[EKMR99] H. Ehrig, H.-J. Kreowski, U. Montanari, G. Rozenberg (eds.). *Handbook of Graph Grammars and Computing by Graph Transformation*. Volume 3: Concurrency, Parallelism, and Distribution. World Scientific, 1999.

[Eve79] S. Even. *Graph Algorithms*. Computer Science Press, Rockville, Maryland, 1979.

[Hab92] A. Habel. *Hyperedge Replacement: Grammars and Languages*. LNCS 643. Springer, Berlin, 1992.

[Har69] F. Harary. *Graph Theory*. Addison-Wesley, Reading, Mass., 1969.

[HESV91] A. Hsu, F. Eskafi, S. Sachs, P. Varaiya. The Design of Platoon Maneuver Protocols for IVHS. Technical report, Institute of Transportation Studies, University of California at Berkeley, 1991.

[HHT96] A. Habel, R. Heckel, G. Taentzer. Graph Grammars with Negative Application Conditions. *Fundamenta Informaticae* 26:287–313, 1996.

[HP05] A. Habel, K.-H. Pennemann. Nested Constraints and Application Conditions for High-Level Structures. In *Formal Methods in Software and System Modeling*. LNCS 3393, pp. 293–308. Springer, 2005.

[HP09] A. Habel, K.-H. Pennemann. Correctness of High-Level Transformation Systems Relative to Nested Conditions. *Mathematical Structures in Computer Science* 19:245–296, 2009.

[HW95] R. Heckel, A. Wagner. Ensuring Consistency of Conditional Graph Grammars—A Constructive Approach. In *SEGRAGRA '95*. ENTCS 2, pp. 95–104. 1995.

[KMP05] M. Koch, L. V. Mancini, F. Parisi-Presicce. Graph-based Specification of Access Control Policies. *Journal of Computer and System Sciences* 71:1–33, 2005.

[LS04] S. Lack, P. Sobociński. Adhesive Categories. In *Foundations of Software Science and Computation Structures (FOSSACS'04)*. LNCS 2987, pp. 273–288. Springer, 2004.

[Pen09] K.-H. Pennemann. *Development of Correct Graph Transformation Systems*. PhD thesis, Universität Oldenburg, 2009. http://formale-sprachen.informatik.uni-oldenburg.de/skript/fs-pub/diss_pennemann.pdf.

[PH96] D. Plump, A. Habel. Graph Unification and Matching. In *Graph Grammars and Their Application to Computer Science*. LNCS 1073, pp. 75–89. Springer, 1996.

[Pra04] U. Prange. Graphs with Variables as an Adhesive HLR Category. Technical report, Technische Universität Berlin, Falultät IV — Elektrotechnik und Informatik, 2004.

[Roz97] G. Rozenberg (ed.). *Handbook of Graph Grammars and Computing by Graph Transformation*. Volume 1: Foundations. World Scientific, 1997.

A Weak adhesive HLR categories

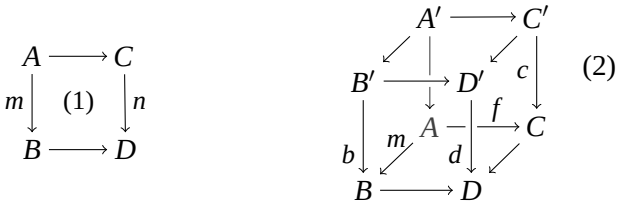
We recall the notions of weak adhesive HLR categories, i.e. categories based on objects of many kinds of structures which are of interest in computer science and mathematics, e.g. Petri-nets, (hyper)graphs, and algebraic specifications, together with their corresponding morphisms and with specific properties. Readers interested in the category-theoretic background of these concepts may consult e.g. [EEPT06b].

Definition 7 (Weak adhesive HLR category) A category $\mathscr{C}$ with a morphism class $\mathscr{M}$ is a *weak adhesive HLR category*, if the following properties hold:

1. *$\mathscr{M}$ is a class of monomorphisms closed under isomorphisms, composition, and decomposition*, i.e., for morphisms f and g, $f \in \mathscr{M}$, g isomorphism (or vice versa) implies $g \circ f \in \mathscr{M}$; $f, g \in \mathscr{M}$ implies $g \circ f \in \mathscr{M}$; and $g \circ f \in \mathscr{M}$, $g \in \mathscr{M}$ implies $f \in \mathscr{M}$.

2. *$\mathscr{C}$ has pushouts and pullbacks along $\mathscr{M}$-morphisms*, i.e. pushouts and pullbacks, where at least one of the given morphisms is in $\mathscr{M}$, and *$\mathscr{M}$-morphisms are closed under pushouts and pullbacks*, i.e. given a pushout (1) as in the figure below, $m \in \mathscr{M}$ implies $n \in \mathscr{M}$ and, given a pullback (1), $n \in \mathscr{M}$ implies $m \in \mathscr{M}$.

3. *Pushouts in $\mathscr{C}$ along $\mathscr{M}$-morphisms are weak VK-squares*, i.e. for any commutative cube (2) in $\mathscr{C}$ with pushout (1) with $m \in \mathscr{M}$ and ($f \in \mathscr{M}$ or $b, c, d \in \mathscr{M}$) in the bottom and where the back faces are pullbacks, the following statement holds: the top face is a pushout iff the front faces are pullbacks.



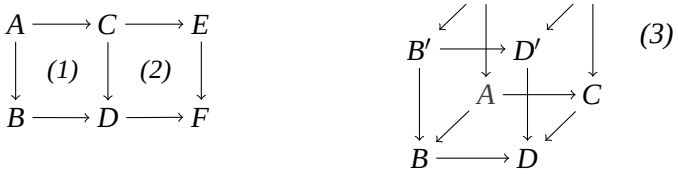
Fact 5 (Graphs is weak adhesive HLR [EEPT06b]) *The category* $\langle \text{Graphs}, \text{Inj} \rangle$ *of graphs with class* Inj *of all injective graph morphisms is a weak adhesive HLR category.*

Further examples of weak adhesive HLR categories are the categories of hypergraphs with all injective hypergraph morphisms, place-transition nets with all injective net morphisms, and algebraic specifications with all strict injective specification morphisms.

Weak adhesive HLR-categories have a number of nice properties, called HLR properties.

Fact 6 (HLR-properties [LS04, EEPT06b]) *For a weak adhesive HLR-category* $\langle \mathscr{C}, \mathscr{M} \rangle$*, the following properties hold:*

1. Pushouts along $\mathscr{M}$-morphisms are pullbacks.
2. $\mathscr{M}$ pushout-pullback decomposition. *If the diagram (1)+(2) in the figure below is a pushout, (2) a pullback,* $D \to F$ *in* $\mathscr{M}$ *and* ($A \to B$ *or* $A \to C$ *in* $\mathscr{M}$*), then (1) and (2) are pushouts and also pullbacks.*
3. Cube pushout-pullback decomposition. *Given the commutative cube (3) in the figure below, where all morphisms in the top and the bottom are in* $\mathscr{M}$*, the top is pullback, and the front faces are pushouts, then the bottom is a pullback iff the back faces of the cube are pushouts.*



4. Uniqueness of pushout complements. *Given morphisms* $A \to C$ *in* $\mathscr{M}$ *and* $C \to D$*, then there is, up to isomorphism, at most one* B *with* $A \to B$ *and* $B \to D$ *such that diagram (1) is a pushout.*

B Constructions and proofs

In this section, show that the category $\langle \text{XGraphs}, \mathscr{M} \rangle$ of graphs with variables with the class $\mathscr{M}$ of all injective graph morphisms is a weak adhesive HLR category. For this purpose, we show that $\mathscr{M}$ is a class of monomorphisms closed under isomorphisms, composition, and decomposition (Lemma 1), the category has pushouts and pullbacks along $\mathscr{M}$-morphisms (Lemma 2), and pushouts along $\mathscr{M}$-morphisms are weak VK squares (Lemma 3).

Lemma 1 *In* XGraphs*, the following properties hold:*

1. *$\mathscr{M}$-morphisms are monomorphisms.*
2. *$\mathscr{M}$-morphisms are closed under composition and decomposition.*

Proof. Straightforward □

In the following, we have to show the existence of pushouts and pullbacks along $\mathscr{M}$-morphisms. Since every morphism consists of a replacement and a graph mophisms, we first give a construction for an injective graph morphism and a replacement. The remainder construction can be done as usual for graph morphisms.

Lemma 2 (Pushouts and pullbacks along $\mathscr{M}$-morphisms) *For every morphism $m\colon A \hookrightarrow B$ in $\mathscr{M}$ and every replacement $\rho\colon A \Rightarrow C$, there is an object D, a morphism $m'\colon C \hookrightarrow D$ in $\mathscr{M}$ and a replacement $\rho'\colon B \Rightarrow D$ forming a pushout. Vice versa, for every morphism $m'\colon C \hookrightarrow D$ in $\mathscr{M}$ and every replacement $\rho'\colon B \Rightarrow D$, there is an object A, a morphism $m\colon A \hookrightarrow B$ in $\mathscr{M}$, and a replacement $\rho'\colon A \Rightarrow C$ and forming a pullback.*

$$\begin{array}{ccc} A & \overset{\rho}{\Longrightarrow} & C \\ {\scriptstyle m}\downarrow & (1) & \downarrow{\scriptstyle m'} \\ B & \underset{\rho'}{\Longrightarrow} & D \end{array}$$

Proof. The pushout object D, m', and ρ' are constructed as follows.

- $\mathrm{V}_D = \mathrm{V}_B + (\mathrm{V}_C - \mathrm{V}_A)$
- $\mathrm{E}_D = \mathrm{E}_B + (\mathrm{E}_C - \mathrm{E}_A)$
- $\mathrm{Y}_D = \mathrm{Y}_B + (\mathrm{Y}_C - \mathrm{Y}_A) - m_{\mathrm{Y}}(\mathrm{Dom}(\rho))$
- $\mathrm{s}_D(e) =$ if $e \in \mathrm{E}_B$ then $\mathrm{s}_B(e)$ else if $e \in \mathrm{E}_C$ and $\mathrm{s}_C(e) \in \mathrm{V}_C - \mathrm{V}_A$ then $\mathrm{s}_C(e)$ else $m_{\mathrm{V}}(\mathrm{s}_C(e))$ for $e \in \mathrm{E}_D$. $\mathrm{t}_D(e)$ and $\mathrm{att}_D(y)$ are defined analogously.
- $\mathrm{lv}_D(v) =$ if $v \in \mathrm{V}_B$ then $\mathrm{lv}_B(v)$ else $\mathrm{lv}_C(v)$ for $v \in \mathrm{V}_D$. $\mathrm{le}_D(e)$ and $\mathrm{ly}_D(y)$ are defined analogously.
- $m'\colon C \to D$ is given by $\langle m'_{\mathrm{V}}, m'_{\mathrm{E}} \rangle$ where m'_{V} is defined by $\mathrm{m}'_{\mathrm{V}}(v) =$ if $v \in \mathrm{V}_A$ then $m_{\mathrm{V}}(v)$ else v for all $v \in \mathrm{V}_C$, and m'_{E} is defined analogously. $\rho'\colon B \Rightarrow D$ is defined by $\rho' = \{m(y)/R \mid y/R \in \rho\}$.

Then $m'\colon C \hookrightarrow D$ is a morphism in $\mathscr{M}$, $\rho'\colon B \Rightarrow D$ is a replacement, and the constructed diagram is a pushout. The pullback object A, m, and ρ are as follows.

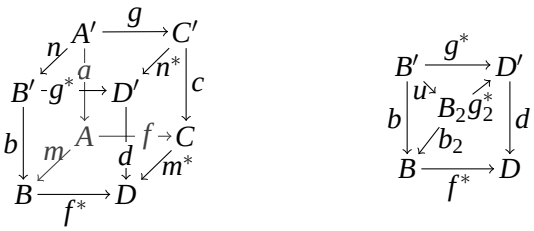
- $\mathrm{V}_A = \mathrm{V}_B \cap \{m'_{\mathrm{V}}(v) \mid v \in \mathrm{V}_C\}$
- $\mathrm{E}_A = \mathrm{E}_B \cap \{m'_{\mathrm{E}}(e) \mid e \in \mathrm{E}_C\}$

- $\mathrm{Y}_A = \mathrm{Y}_B \cap \{m'_\mathrm{Y}(y) \mid y \in \mathrm{Y}_C\} \cup \mathrm{Y}_C$
- $\mathrm{s}_A(e) = \mathrm{s}_B(e)$ for all $e \in \mathrm{E}_A$. $\mathrm{t}_A(e)$ is defined analogously.
  $\mathrm{att}_A(y) =$ if $y \in C$ then $\mathrm{att}_C(y)$ else $\mathrm{att}_B(y)$ for all $y \in \mathrm{Y}_A$.
- $\mathrm{lv}_A(v) = \mathrm{lv}_B(v)$ for all $v \in \mathrm{V}_A$. $\mathrm{le}_A(e)$ and $\mathrm{ly}_A(y)$ are defined analogously.
- $m\colon A \to B$ is restriction of $m'\colon C \to D$ to A and $\rho\colon A \to C$ is defined by $\rho = \{y/R \mid y \in \mathrm{Y}_A, (m_\mathrm{Y}(y)/R) \in \rho'\}$.

Then $m\colon A \to B$ a graph morphism in $\mathscr{M}$, $\rho\colon A \Rightarrow C$ is a replacement, and the constructed diagram is a pullback. □

Lemma 3 *In* XGraphs, *pushouts along $\mathscr{M}$-morphisms are weak VK squares.*

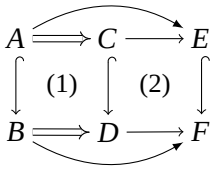
Proof. By case analysis, similar to the proof in [EEPT06a]. In the following, we will use the notation *ABCD* to designate the square from $B \leftarrow A \rightarrow C$ to $B \rightarrow D \leftarrow C$. In the commutative cube below, let *ABCD* be a pushout with $m \in \mathscr{M}$ and the back faces $A'AB'B$ and $A'AC'C$ be pullbacks. Assume that $f \in \mathscr{M}$ or $b, c, d \in \mathscr{M}$.



We proceed by case distinction.
Case 1. $f \in \mathscr{M}$ and one of the back arrows, i.e. a, b, c, is in $\mathscr{M}$. By Lemma 2, all morphisms except the "vertical" morphisms a, b, c, d are in $\mathscr{M}$. Because the back sides of the cube are pullbacks, all of a, b, c are in $\mathscr{M}$. By commutativity of the cube, $d \in \mathscr{M}$. The proof proceeds as for the category of hypergraphs in [EEPT06a].
Case 2. $b, c, d \in \mathscr{M}$ and one of g, f, f^* is in $\mathscr{M}$. Analogous to Case 1.
Case 3. f is in $\mathscr{M}$ and none of the back arrows a, b, c are in $\mathscr{M}$. Because the back sides are pullbacks, $f^*, g \in \mathscr{M}$.
Case 3.1. Assume the top is a pushout. Then $g, g^* \in \mathscr{M}$. Assume a pullback object B_2 with morphisms $g_2^*\colon B_2 \to D'$ and $b_2\colon B_2 \to B$. Then, there is a unique $u : B' \to B$ with $g^* = g_2^* u$ and $b = b_2 u$. Similar to [EEPT06a], we can show that u is bijective and therefore $B' \cong B_2$.
Case 4. $b, c, d \in \mathscr{M}$ and $g, f, f^* \notin \mathscr{M}$. The proof is analogous to Case 3. □

Proof of Theorem 3. 1.$\mathscr{M}$ is a class of monomorphisms closed under isomorphisms, composition and decomposition: The class of all injective graph morphisms in XGraphs is a class of monomorphisms. It suffices, then, to show that $g \circ f$ is in $\mathscr{M}$ for morphisms $g, f \in \mathscr{M}$, and that $g \circ f \in \mathscr{M}$ implies $f \in \mathscr{M}$. Both propositions are true due to [AHS90, Proposition 7.34].
2. XGraphs has pushouts and pullbacks along $\mathscr{M}$-morphisms: For $A \hookrightarrow B$ in $\mathscr{M}$, the pushout of $B \hookleftarrow A \rightarrow E$ is constructed by splitting the morphism $A \rightarrow E$ into a replacement $\rho\colon A \Rightarrow C$ and a morphism $C \rightarrow E$, constructing (1) as pushout of $B \hookleftarrow A \Rightarrow C$ accoring to Lemma 2 and (2) as

pushout of $D \hookleftarrow C \to E$ as usual (e.g. in the category of hypergraphs [EEPT06b]). Then (1)+(2) is a pushout.



For $E \hookrightarrow F$ in $\mathscr{M}$, the pullback of $B \hookrightarrow F \leftarrow E$ is constructed by splitting the morphism $B \hookrightarrow F$ into a replacement $\rho : B \Rightarrow D$ and a morphism $D \to F$, constructing (2) as usual as pullback (e.g., as in the category of hypergraphs [EEPT06b]) and (1) as pullback according to Lemma 2. Then (1)+(2) is a pullback. □