URL: http://www.elsevier.nl/locate/entcs/volume51.html 12 pages

Basic Results for Two Types of High-Level Replacement Systems ⋆

Hartmut Ehrig [1]

Fachbereich Informatik
Technische Universität Berlin
D-10587 Berlin, Germany

Annegret Habel [2]

Fachbereich Informatik
Universität Oldenburg
Postfach 2503, D-26111 Oldenburg, Germany

F. Parisi-Presicce [3]

Universitá di Roma "La Sapienza"
Dipartimento Scienze dell'Informazione
I-00198 Roma, Italy

Abstract

The general idea of high-level replacement systems is to generalize the concept of graph transformation systems and graph grammars from graphs to all kinds of structures which are of interest in Computer Science and Mathematics. Within the algebraic approach of graph transformation this is possible by replacing graphs, graph morphisms, and pushouts (gluing) of graphs by objects, morphisms, and pushouts in a suitable category. Of special interest are categories for all kinds of labelled and typed graphs, hypergraphs, algebraic specifications and Petri nets. In this paper, we review the basic results for high-level replacement systems in the algebraic double-pushout approach in the symmetric case, where both rule morphisms belong to a distinguished class $\mathcal{M}$. Moreover we present for the first time the asymmetric type of high-level replacement systems, where only the left rule morphism $K \to L$ belongs to $\mathcal{M}$.

⋆ Research partially supported by the European Community under TMR Network GETGRATS and the ESPRIT Working Group APPLIGRAPH.

[1] Email: ehrig@cs.tu-berlin.de
[2] Email: habel@informatik.uni-oldenburg.de
[3] Email: parisi@dsi.uniroma1.it

1 Introduction

The general idea of high-level replacement systems is to generalize the concepts of graph replacement form graphs to all kinds of structures which are of interest in Computer Science and Mathematics (see [3,4,5] for more details). In this paper we review the basic concepts and main results in the double-pushout approach. This generalizes corresponding concepts and results in the algebraic theory of graph grammars (see [1,2]) and can also be applied to algebraic specifications and Petri nets (see [3,7,11]). The algebraic theory of graph grammars has been revisited in [8] for asymmetric rules, where only the left-hand side is injective. This type of graph grammars is generalized in this paper to high-level replacement systems of type DPO$'$ and suitable HLR$'$-conditions are formulated in order to be able to show the basic results concerning Church-Rosser properties and parallelism. These basic results are the basis for the algebraic theory of graph grammars (see [1,2]), where especially problems of abstract semantics, parallelism and concurrency are studied. Part of these more advanced results have been achieved already for type DPO (see [3,4,5]), but it remains open to study them for type DPO$'$.

2 Basic HLR-concepts

In this section we review the basic concepts of high-level replacement systems (HLR-systems) including productions, derivations and systems. In what follows, **CAT** is a category with a distinguished class $\mathcal{M}$ of morphisms.

Definition 2.1 (Rules and Transformations) *A* rule $p = (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ *in* **CAT** *consists of three objects L, K and R, called left-hand side, interface (or gluing object), and right-hand side, respectively, and two morphisms $K \stackrel{l}{\rightarrow} L$ and $K \stackrel{r}{\rightarrow} R$ with the morphism $l \in \mathcal{M}$.*

Given a rule $p = (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ a direct transformation $G \stackrel{p}{\Longrightarrow} H$*, from an object G to an object H is given by the following two pushout diagrams (1) and (2) in the category* **CAT***:*

$$\begin{tikzcd} L \arrow[d, "g"'] \arrow[dr, phantom, "(1)"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "c"] \arrow[dr, phantom, "(2)"] & R \arrow[d, "h"] \\ G & C \arrow[l, "l^*"] \arrow[r, "r^*"'] & H \end{tikzcd}$$

The morphisms $L \stackrel{g}{\rightarrow} G$ and $R \stackrel{h}{\rightarrow} H$ are called occurrences of L in G and R in H, respectively.

The existence of an occurrence of L in G is not sufficient for the applicability of p. In order to apply a rule to a given object, a gluing condition has to be satisfied (see [2]). In our abstract framework, the gluing condition is satisfied if there exists an object C such that the given object G becomes a pushout object.

Fact 2.2 (Applicability of Rules) *Given a rule $p = (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)$, an object G and an occurrence $L \overset{g}{\rightarrow} G$ of L in G, then the rule p is applicable to G via $L \overset{g}{\rightarrow} G$ if the following two conditions are satisfied:*

(i) *There is an object C (called the pushout complement object) together with morphisms $K \overset{c}{\rightarrow} C$ and $C \overset{l^*}{\rightarrow} G$, such that the square (1) in Definition 2.1 is a pushout square.*

(ii) *There is an object H together with morphisms $R \overset{h}{\rightarrow} H$ and $C \overset{r^*}{\rightarrow} H$, such that the square (2) in Definition 2.1 is a pushout square.*

If both conditions are satisfied, a direct transformation $G \overset{p}{\Longrightarrow} H$ can be constructed. It is unique up to isomorphism if and only if the pushout complement construction is unique. Given a concrete category, the gluing condition can be given in a constructive way, for graphs see [2], for algebraic specifications see [3], for algebraic high-level nets see [11].

Now we are able to define high-level replacement systems in an arbitrary category generalizing the concept of graph grammars in the double-pushout approach. The initial graph is replaced by a start object and the set of rules consisting of a pair of injective graph morphisms is generalized by a set of rules with morphisms in a distinguished class $\mathcal{M}$.

Definition 2.3 (High-Level Replacement System) *Given a category* **CAT** *together with a distinguished class of morphisms $\mathcal{M}$, a high-level replacement system $H = (\mathcal{S}, \mathcal{P})$ in $(\mathbf{CAT}, \mathcal{M})$ of type* DPO′ *is given by a start object $\mathcal{S} \in |\mathbf{CAT}|$, and a set of rules $\mathcal{P}$. The system H is said to be of type* DPO *if for all $p = (L \overset{l}{\leftarrow} C \overset{r}{\rightarrow} R) \in \mathcal{P}$ both morphisms l and r belong to $\mathcal{M}$.*

Example 2.4 (Special Categories) (i) *If* **CAT** *is the category* **Gra** *of directed graphs and graph morphisms as considered in [1] and $\mathcal{M}$ the class of injective graph morphisms, then a high-level replacement system in (***Gra***, $\mathcal{M}$) of type* DPO *(resp.* DPO′*) is a graph grammar in the double-pushout approach as studied in [1] (resp. [6,9,8]).*

(ii) *By choosing in the category* **Set** *of sets and (total) functions the class $\mathcal{M}$ of injective functions, we obtain high-level replacement systems in (***Set***, $\mathcal{M}$) where the transformed high-level structures are sets. This case is of interest, because graphs and several other high-level structures are based on sets in each component.*

(iii) *Taking* **CAT** *to be the category* **Spec** *of algebraic specifications with suitable classes $\mathcal{M}$ of injective specification morphisms leads to algebraic specification grammars in the sense of [7].*

(iv) *Taking* **CAT** *to be the category* **P/T** *of place/transition nets in the sense of [10], where, however, the homomorphism $f_P^{\oplus}: P_1^{\oplus} \rightarrow P_2^{\oplus}$ is generated by a function $f_P: P_1 \rightarrow P_2$ (see [3]), we obtain transformation systems for Petri nets.*

(v) *Taking* **CAT** *to be the category* **Ahl** *of algebraic high-level nets leads to algebraic high-level net transformation systems in the sense of [11].*

3 HLR-concepts for Independence and Parallelism

In this section we formulate the notions of sequential and parallel independence of transformations for high-level replacement systems of type DPO and DPO′, and we present two Church-Rosser theorems and the Parallelism theorem for high-level replacement systems, which are well-known in the case of graph grammars [6,9,2,8]. For most of the proofs, however, we need additional conditions, called HLR- and HLR′-conditions for systems of type DPO and DPO′, respectively, for the category **CAT**. These conditions will be formulated in section 4.

First of all we need categorical formulations of sequential and parallel independence which are required in all these results. The intuitive meaning of sequential independence of two direct transformations $G \overset{p_1}{\Longrightarrow} H$ and $H \overset{p_2}{\Longrightarrow} X$ via the rules $p_i = (L_i \leftarrow K_i \rightarrow R_i)$ for $i = 1, 2$ is that the intersection of R_1 and L_2 in H is included in the intersection of K_1 and K_2 in H. In other words, the first rule deletes nothing that is needed by the second rule and the second rule does not need anything produced by the first rule. In the case of high-level-replacement systems of type DPO, this is equivalent to the existence of suitable morphisms $R_1 \rightarrow C_2$ and $L_2 \rightarrow C_1$ as stated in the following definition of sequential independence (see [1]). The formulation for parallel independence is similar.

Definition 3.1 (Sequential Independence) *Given two direct transformations $G \overset{p_1}{\Longrightarrow} H$ and $H \overset{p_2}{\Longrightarrow} X$ as in the diagram:*

$$
\begin{tikzcd}
L_1 \arrow[d] & K_1 \arrow[l] \arrow[r] \arrow[d] & R_1 \arrow[dr] \arrow[drrr, dashed] & & L_2 \arrow[dl] \arrow[dlll, dashed] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
G & C_1 \arrow[l] \arrow[rr] \arrow[r, phantom, shift left=3, "="] & & H & & C_2 \arrow[ll] \arrow[r] \arrow[l, phantom, shift left=3, "="] & X
\end{tikzcd}
$$

the transformations $G \overset{p_1}{\Longrightarrow} H$ and $H \overset{p_2}{\Longrightarrow} X$ are called sequentially independent *if there are morphisms $R_1 \rightarrow C_2$ and $L_2 \rightarrow C_1$, so that $R_1 \rightarrow C_2 \rightarrow H = R_1 \rightarrow H$ and $L_2 \rightarrow C_1 \rightarrow H = L_2 \rightarrow H$.*

Definition 3.2 (Parallel Independence) *Given two direct transformations $G \overset{p_1}{\Longrightarrow} H_1$ and $G \overset{p_2}{\Longrightarrow} H_2$ as in the diagram:*

$$
\begin{tikzcd}
R_1 \arrow[d] & K_1 \arrow[l] \arrow[r] \arrow[d] & L_1 \arrow[dr] \arrow[drrr, dashed] & & L_2 \arrow[dl] \arrow[dlll, dashed] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
H_1 & C_1 \arrow[l] \arrow[rr] \arrow[r, phantom, shift left=3, "="] & & G & & C_2 \arrow[ll] \arrow[r] \arrow[l, phantom, shift left=3, "="] & H_2
\end{tikzcd}
$$

the transformations $G \overset{p_1}{\Longrightarrow} H_1$ and $G \overset{p_2}{\Longrightarrow} H_2$ are called parallel independent *if there are morphisms $L_1 \rightarrow C_2$ and $L_2 \rightarrow C_1$, so that $L_1 \rightarrow C_2 \rightarrow G = L_1 \rightarrow G$*

and $L_2 \to C_1 \to G = L_2 \to G$.

For the following local Church-Rosser and parallelism theorems we require that a high-level replacement system of type DPO satisfies the HLR-conditions and of type DPO$'$ satisfies the HLR$'$-conditions stated in section 4, where also the main ideas for the proofs are given. Note, that for the DPO$'$-type we don't have to require that the vertical morphisms are in $\mathcal{M}$ as required in some of the graph cases in [8].

Theorem 3.3 (Local Church-Rosser I) *Given parallel independent direct transformations* $G \overset{p_1}{\Longrightarrow} H_1$ *and* $G \overset{p_2}{\Longrightarrow} H_2$ *, there is an object* X *and direct transformations* $H_1 \overset{p_2}{\Longrightarrow} X$ *and* $H_2 \overset{p_1}{\Longrightarrow} X$*, so that the transformations* $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ *and* $G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$ *are sequentially independent.*

Theorem 3.4 (Local Church-Rosser II) *Given a sequentially independent transformation* $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ *there exist an object* H_2 *and sequentially independent transformations* $G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$ *so that the transformations* $G \overset{p_1}{\Longrightarrow} H_1$ *and* $G \overset{p_2}{\Longrightarrow} H_2$ *are parallel independent.*

If the category **CAT** has binary coproducts, denoted by $+$ we are able to formulate parallel productions and transformations.

Definition 3.5 (Parallel Rule) *Given rules* $p_1 = (L_1 \overset{l_1}{\leftarrow} K_1 \overset{r_1}{\rightarrow} R_1)$ *and* $p_2 = (L_2 \overset{l_2}{\leftarrow} K_2 \overset{r_2}{\rightarrow} R_2)$*, the rule* $p_1 + p_2 = (L_1 + L_2 \overset{l_1+l_2}{\leftarrow} K_1 + K_2 \overset{r_1+r_2}{\rightarrow} R_1 + R_2)$ *defined by binary coproducts in* **CAT** *is called the* parallel rule *of* p_1 *and* p_2*. A transformation* $t\colon G \overset{p_1+p_2}{\Longrightarrow} X$ *defined by parallel rules is called a parallel transformation.*

Theorem 3.6 (Parallelism) *If* p_1 *and* p_2 *are rules and* $p_1 + p_2$ *is the corresponding parallel rule, then:*

(i) **Synthesis:**
Given a sequentially independent transformation $s\colon G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$*, there is a parallel transformation* $t\colon G \overset{p_1+p_2}{\Longrightarrow} X$.

(ii) **Analysis:**
Given a parallel transformation $t\colon G \overset{p_1+p_2}{\Longrightarrow} X$*, there are two sequentially independent transformations* $s_1\colon G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ *and* $s_2\colon G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$.

(iii) **Bijective correspondence:**
There is a bijective correspondence between sequentially independent and parallel transformations, that is, given a sequentially independent transformation s *the "synthesis" construction leads to a parallel transformation* t *and the "analysis" construction leads back to the same sequentially independent transformation* s *(up to isomorphism), and vice versa.*

4 HLR- and HLR′-Conditions

In the following we first review the conditions, called HLR-conditions, which are required in the proofs of the Church-Rosser Theorem I and II and the Parallelism Theorem for high-level replacement systems of type DPO.

Definition 4.1 (HLR-**Conditions**) *Given a category* **CAT** *and a distinguished class* $\mathcal{M}$ *of morphisms in* **CAT***, the following conditions are called* HLR-*conditions for* $(\mathbf{CAT}, \mathcal{M})$*:*

(1) Existence of semi-$\mathcal{M}$ pushouts.
For all objects A,B,C and morphisms $C \leftarrow A \rightarrow B$, where at least one is in $\mathcal{M}$, there exists a pushout $C \rightarrow D \leftarrow B$:

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr, phantom, "(1)"] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd}$$

(2) Existence of $\mathcal{M}$-pullbacks.
For all morphisms $B \rightarrow D$ and $C \rightarrow D$ both in $\mathcal{M}$, there exists a pullback $C \leftarrow A \rightarrow B$ as in diagram (1) above.

(3) Inheritance of $\mathcal{M}$-morphisms under pushouts and $\mathcal{M}$-pullbacks.
 (a) For each pushout square (1) as above $A \rightarrow B \in \mathcal{M}$ implies $C \rightarrow D \in \mathcal{M}$.
 (b) For each pullback square (1) as above $B \rightarrow D \in \mathcal{M}$ and $C \rightarrow D \in \mathcal{M}$ implies $A \rightarrow B \in \mathcal{M}$ and $A \rightarrow C \in \mathcal{M}$.

(4) $\mathcal{M}$-pushout-pullback-decomposition.
For each diagram of the form:

$$\begin{tikzcd} A \arrow[r] \arrow[d] \arrow[dr, phantom, "(1)"] & B \arrow[r] \arrow[d] \arrow[dr, phantom, "(2)"] & E \arrow[d] \\ C \arrow[r] & D \arrow[r] & F \end{tikzcd}$$

If (1+2) is a pushout square, (2) is a pullback square and $A \rightarrow C$, $B \rightarrow D$, $E \rightarrow F$, $B \rightarrow E$ and $D \rightarrow F$ are $\mathcal{M}$-morphisms, then also (1) is a pushout square.

(5) Existence of binary coproducts and compatibility with $\mathcal{M}$.
 (a) For each pair of objects A, B there is a coproduct $A + B$ with the universal morphisms $A \rightarrow A + B$ and $B \rightarrow A + B$.
 (b) For each pair of morphisms $A \xrightarrow{f} A'$ and $B \xrightarrow{g} B'$ in $\mathcal{M}$ the coproduct morphism $A + B \xrightarrow{f+g} A' + B'$ is also in $\mathcal{M}$.

(6) $\mathcal{M}$-pushouts are pullbacks.
Pushout squares of $\mathcal{M}$-morphisms are pullback squares.

Note that variants of these HLR-conditions have been stated in [4,5] in order to prove local Church-Rosser and Parallelism Theorems in the framework of high-level replacement systems. In fact, the conditions above imply those in [4,5], where they are called "HLR1" conditions.

Proof Ideas of Main Results for Type DPO **with** HLR**-Conditions.** The proofs of Theorems 3.3, 3.4 and 3.6 for high-level replacement systems of type DPO can be found in [5], page 380, page 377 and page 382, respectively. In particular, the proofs of the local Church-Rosser Theorem I and II make use of (1) the existence of semi-$\mathcal{M}$ pushouts, (2) the existence of $\mathcal{M}$ pullbacks, (3) inheritance of $\mathcal{M}$-morphisms under pushouts and $\mathcal{M}$-pullbacks and (4) $\mathcal{M}$-pushout-pullback-decomposition. Condition (5) Existence of binary coproducts and compatibility with $\mathcal{M}$ ensures that for each pair (p_1, p_2) of productions there is a parallel production $p_1 + p_2$ which is again a production in the sense of 2.1. Finally, statement (i) and (ii) in the Parallelism Theorem hold provided that the DPO-conditions (1)–(5) are satisfied, and the Bijective Correspondence (iii) holds provided that additionally, the DPO-condition (6) $\mathcal{M}$-pushouts are pullbacks is satisfied.

If only the left morphism $K \to L$ of a rule p is in $\mathcal{M}$ and the high level replacement system is of type DPO$'$, then different conditions, called HLR$'$-conditions, are needed in the proofs of the local Church-Rosser and parallelism theorems.

Definition 4.2 (HLR$'$**-Conditions)** *Given a category* **CAT** *(of high-level structures) and a distinguished class $\mathcal{M}$ of morphisms in* **CAT** *the following conditions are called* HLR$'$*-conditions for* $(\mathbf{CAT}, \mathcal{M})$*:*

(1') Existence of arbitrary pushouts.
For all objects A,B,C and morphisms $C \leftarrow A \to B$ there exists a pushout $C \to D \leftarrow B$.

(2') Existence of semi-$\mathcal{M}$-pullbacks.
For all morphisms $B \to D$ and $C \to D$ where at least one is in $\mathcal{M}$, there exists a pullback $C \leftarrow A \to B$ as in diagram (1) above.

(3') Inheritance of $\mathcal{M}$-morphisms under pushouts and pullbacks.
- *(a) For each pushout square (1) as above $A \to B \in \mathcal{M}$ implies $C \to D \in \mathcal{M}$.*
- *(b') For each pullback square (1) as above $C \to D \in \mathcal{M}$ implies $A \to B \in \mathcal{M}$.*

(4') Semi-$\mathcal{M}$-pushout-pullback-decomposition.
For each diagram (1+2) as above we have:
- *(a) If (1+2) is a pushout square, (2) is a pullback square and $B \to E$ and $D \to F$ are $\mathcal{M}$-morphisms, then also (1) is a pushout square.*
- *(b) If (1+2) is a pushout square, (2) is a pullback and pushout square and $A \to C$, $B \to D$ and $E \to F$ are $\mathcal{M}$-morphisms, then also (1) is a pushout square.*

(5) Existence of binary coproducts and compatibility with $\mathcal{M}$.

(6') Semi-$\mathcal{M}$-pushouts are pullbacks.

Pushout squares of morphisms, where at least one is in $\mathcal{M}$, are pullback squares.

For high-level replacement systems of type DPO$'$ the Church-Rosser Theorem I and II and the Parallelism Theorem hold, provided that the HLR$'$-conditions are satisfied. The proofs of the Church-Rosser Theorem I and the Parallelism Theorem follow by inspecting the proofs in [5], page 380 and page 382, respectively. For HLR-systems of type DPO$'$ we need a separate proof of local Church-Rosser Theorem II which follows for type DPO by symmetry from local Church Rosser Theorem I.

Proof Ideas of Main Results for Type DPO$'$ with HLR$'$-Conditions. The proofs of Theorems 3.3 and 3.6 follow by inspecting the proofs in [5], page 380 and page 382, respectively; the proof of Theorem 3.4 is similar to the one for Theorem 3.3, but requires stronger conditions: The proof of the local Church-Rosser Theorem I makes use of (1') the existence of pushouts, (2) the existence of $\mathcal{M}$ pullbacks, (3) inheritance of $\mathcal{M}$-morphisms under pushouts and $\mathcal{M}$-pullbacks and (4) $\mathcal{M}$-pushout-pullback-decomposition. The proof the local Church-Rosser Theorem II is similar to the one of the local Church-Rosser Theorem I, but it requires the (1') the existence of pushouts, (2') the existence of semi-$\mathcal{M}$ pullbacks, (3') inheritance of $\mathcal{M}$-morphisms under pushouts and pullbacks and (4') semi-$\mathcal{M}$-pushout-pullback-decomposition. Condition (5) Existence of binary coproducts and compatibility with $\mathcal{M}$ ensures that for each pair (p_1, p_2) of productions there is a parallel production $p_1 + p_2$ which is again a production in the sense of 2.1. Finally, statement (i) and (ii) in the Parallelism Theorem hold provided that the DPO$'$-conditions (1')–(4') and (5) are satisfied, and the Bijective Correspondence (iii) holds provided that additionally the DPO$'$-condition (6') semi-$\mathcal{M}$-pushouts are pullbacks is satisfied.

Note that for high-level replacement systems of type DPO$'$ the Synthesis step may yield several distinct parallel transformations $G \overset{p_1+p_2}{\Longrightarrow} X$ via different occurrences of $L_1 + L_2$ in G because for sequentially independent transformations $G \overset{p_1}{\Longrightarrow} H \overset{p_2}{\Longrightarrow} X$ as in Definition 3.1 there may exist several morphisms $L_2 \to C_1$, so that $L_2 \to C_1 \to H = L_2 \to H$ (see Example 6.9 in [8]).

Fact 4.3 (Categories satisfying HLR- (resp. HLR$'$-) Conditions) *Given a category* **CAT** *and a distinguished class $\mathcal{M}$ of morphisms in* **CAT**, *the* HLR$'$-*conditions for* $(\mathbf{CAT}, \mathcal{M})$ *imply the* HLR-*conditions. In particular, the* HLR- *and* HLR$'$-*conditions are satisfied for the following categories* **CAT** *and distinguished classes of morphisms $\mathcal{M}$ discussed in Example 2.4 and in Fact 4.8. $\mathcal{M}_{injective}$ denotes the class of all injective morphisms in the corresponding category and the index* "strict" *is explained below (see also [3] Section 6.3):*

(i) $(\mathbf{Gra}, \mathcal{M}_{injective})$,

(ii) $(\mathbf{Set}, \mathcal{M}_{injective})$,

(iii) $(\mathbf{Spec}, \mathcal{M}_{strict,injective})$,

(iv) $(\mathbf{P/T}, \mathcal{M}_{injective})$,

(v) $(\mathbf{Ahl}, \mathcal{M}_{injective,strict,isom})$,

(vi) $(\mathbf{UGra}, \mathcal{M}_{injective})$.

Proof. The proofs for type DPO with HLR-conditions are given in [3,4,5,11], some time with slightly different notation. The proofs for the categories (i), (ii), and (iii) for type DPO$'$ with HLR$'$-conditions is given below, where the proof for (iv) and (v) is based on that for (i), (ii), and (iii) in the corresponding components. The proof for the category (vi) is given at the end of this section.

In the categories **Set** and **Gra** pushouts, pullbacks, and coproducts exist and their constructions is well-known. This implies HLR$'$-conditions (1'), (2') and (5a) where the last two require those constructions for specific cases only. Inheritance of injective morphisms under pushouts and pullbacks as well as coproducts is well-known for **Set** and implies conditions (3') and (5b). Moreover, it is easy to check explicitly the satisfaction of conditions (4') and (6') for **Set** and injective functions. This implies, in turn, the corresponding conditions for **Gra** because a graph morphism is injective and diagrams are pushouts or pullbacks in **Gra** if and only if the edge and node components of a graph morphism satisfy the corresponding property in **Set**. The proofs for case (iii) are similar and are based on the fact that specification morphisms consist of two set funcions (between the sets of sorts and between the sets of operators) satisfying additional "structural" constraints. More formally, an *algebraic specification morphism* $f: SPEC_1 \to SPEC_2$ between algebraic specifications $SPEC_1 = (\Sigma_1, E_1)$ and $SPEC_2 = (\Sigma_2, E_2)$ is a signature morphism $f_\Sigma: \Sigma_1 \to \Sigma_2$ such that $f^\#_\Sigma(e) \in E_2$ for all $e \in E_1$, i. e., the translation of the equations in E_1 is in E_2. It is *strict* if, in addition, $(f^\#_\Sigma)^{-1}(E_2) \subseteq E_1$, i. e., if any equation in $SPEC_2$ formed only with operation symbols in the image by f_Σ of Σ_1 is the translation of an equation already in $SPEC_1$. As for the cases of **Set** and **Gra**, the HLR$'$-conditions (1'), (2') and (5a) are already satisfied by the known construction of pushouts, pullbacks, and coproducts of algebraic specifications. Conditions (4') and (6') follow from the corresponding properties in **Set** since they are based on injectivity and not on the strictness of the morphisms. Finally, for (3'a), consider the pushout square (1) above. If D contains an equation built only from (the image of) operators of C, by the construction of pushouts in **SPEC**, the equation, if not (the image of an equation) in C, could only be (the image of an equation) from B, hence using only operators from B. But these operators could be common to B and C only if (the image of operators) originally in A. But if $A \to B$ is strict, any equation in B built only with (the image of) operators from A must already be (the translation of an equation) in A, hence in C. The remaining properties can be proved in a similar way.

We conclude the section with one more example of structure giving rise to a high-level replacement system of type DPO′ that satisfies the HLR′-conditions. The framework is that of undirected graphs, where each edge is associated with a set of 1 or 2 nodes, its endpoints.

Definition 4.4 (Undirected Graph) *An* undirected graph (U-graph) G *is a triple* (G_E, G_N, end), *where* G_E *and* G_N, *are the set of edges and the set of nodes, respectively, and* $end: G_E \to \mathbf{P_2}(G_N)$ *is the function associating each edge* $\mathbf{e}$ *to a subset* $end(\mathbf{e})$ *of* G_N *of cardinality* 1 *or* 2.

As for the directed case, morphisms between undirected graphs must preserve the structure.

Definition 4.5 (U-graph morphism) *Given two undirected graphs* $G = (G_E, G_N, end)$, *and* $G' = (G'_E, G'_N, end')$, *a* U-graph morphism $f: G \to G'$ *is a pair* $(f_E: G_N \to G'_N, f_E: G_E \to G'_E)$ *such that* $end'(f_E(e)) = f_N(end(e))$ *(where the same notation is used to denote the obvious extension of* f_N *to subsets of* N*).*

Composition of U-graph morphisms is defined componentwise. Composition is clearly associative and the pair $(id_E: G_N \to G_N, id_E: G_E \to G_E)$ is the obvious identity.

Fact 4.6 *The category* **UGra** *whose objects are the undirected graphs and whose morphisms are the U-graph morphisms is closed under pushouts and pullbacks.*

For undirected graphs, it is easy to adapt the original Gluing Conditions [2] to guarantee the applicability of rules.

Proposition 4.7 *Given* $p = (L \leftarrow K \to R)$ *and* $g: L \to G$, *let*

$$ID_g = \{x \in L: \exists y \in L, x \neq y, g(x) = g(y)\},$$
$$DANG_g = \{n \in L_N: \exists e \in G_E - g_E(L_E) \text{ such that } g_N(n) \in end_G(e)\}.$$

Then the pushout complement C *exists if and only if* $DANG_g \cup ID_g \subseteq l(K)$.

An *injective* U-graph morphism is just a U-graph morphism where both components f_N and f_E are injective functions. With arguments similar to those used for directed graphs [5], it can be shown that (**UGra** gives rise to a high-level replacement system satisfying the HLR′-conditions.

Fact 4.8 *The category* **UGra** *with distinguished class* $\mathcal{M}_{injective}$ *of morphisms forms a high-level replacement system of type* DPO′ *that satisfyies the* HLR′-*conditions.*

Proof. The construction of pushouts, pullbacks and coproducts for arbitrary pairs of U-graph morphisms is based on the corresponding construction of the underlying set functions in **Set** and thus the satisfaction of the structural properties in the HLR′-conditions follows from the fact that $(\mathbf{Set}, \mathcal{M}_{injective})$ satisfies the HLR′-conditions. Since a U-graph morphism is injective exactly when the edge and node components are, the inheritance (3') and the compatibility (5b) of $\mathcal{M}$ morphisms follows from the corresponding properties of **Set** as well.

5 Conclusion

In this paper we have reviewed how to achieve Church-Rosser and parallelism results for HLR-systems in the double pushout approach, short HLR-systems of type DPO. For HLR-systems of type DPO$'$, where only the left-hand side of a rule belongs to a distinguished class $\mathcal{M}$, we have presented corresponding results for the first time. In fact, we have presented HLR$'$-conditions sufficient for type DPO$'$, which are slightly stronger than the HLR-conditions used for type DPO. All our example categories, however, satisfy not only the HLR-, but also the HLR$'$-conditions. Hence it remains open whether there are interesting examples satisfying the HLR-, but not the HLR$'$-conditions.

References

[1] Corradini, A., 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: G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph transformation, Volume 1: Foundations*, World Scientific, 1997 pp. 163–246.

[2] Ehrig, H., *Introduction to the algebraic theory of graph grammars*, in: V. Claus, H. Ehrig and G. Rozenberg, editors, *Lecture Notes in Computer Science 73, 1st Graph Grammar Workshop* (1979), pp. 1–69.

[3] Ehrig, H., M. Gajewsky and F. Parisi-Presicce, *High-level replacement systems applied to algebraic specifications and petri nets*, in: *Handbook of Graph Grammars and Computing by Graph transformation, Volume 3: Concurrency, Parallelism, and Distribution*, World Scientific, 1999 pp. 341–399.

[4] Ehrig, H., A. Habel, H.-J. Kreowski and F. Parisi-Presicce, *From graph grammars to high level replacement systems*, in: *Lecture Notes in Computer Science 532, 4th Int. Workshop on Graph Grammars and their Application to Computer Science* (1991), pp. 269–291.

[5] Ehrig, H., A. Habel, H.-J. Kreowski and F. Parisi-Presicce, *Parallelism and concurrency in high level replacement systems*, Mathematical Structures in Computer Science **1** (1991), pp. 361–404.

[6] Ehrig, H. and H.-J. Kreowski, *Parallelism of manipulations in multidimensional information structures*, in: *Lecture Notes in Computer Science 45, Mathematical Foundations of Computer Science* (1976), pp. 284–293.

[7] Ehrig, H. and F. Parisi-Presicce, *Algebraic specification grammars: a junction between module specifications and graph grammars*, in: *Lecture Notes in*

Computer Science 532, 4th Int. Workshop on Graph Grammars and their Application to Computer Science (1991), pp. 292–310.

[8] Habel, A., J. Müller and D. Plump, *Double-pushout graph transformation revisited*, Mathematical Structures in Computer Science **11** (2001), to appear.

[9] Kreowski, H.-J., *Manipulationen von Graphmanipulationen*, Dissertation, Technische Universität Berlin (1977).

[10] Meseguer, J. and U. Montanari, *Petri nets are monoids*, Information and Computation **88** (1990), pp. 105–155.

[11] Padberg, J., H. Ehrig and L. Ribeiro, *Algebraic high-level net transformation systems*, Mathematical Structures in Computer Science **5** (1995), pp. 217–256.