Double-Pushout Approach with Injective Matching*

Annegret Habel[1], Jürgen Müller[2], and Detlef Plump[3]

[1] Universität Oldenburg, Fachbereich Informatik
Postfach 2503, 26111 Oldenburg, Germany
Annegret.Habel@informatik.uni-oldenburg.de

[2] Universitá di Pisa, Dipartimento di Informatica
Corso Italia 40, 56125 Pisa, Italy.
jmueller@di.unipi.it

[3] Universität Bremen, Fachbereich Mathematik und Informatik
Postfach 33 04 40, 28334 Bremen, Germany
det@informatik.uni-bremen.de

Abstract. We investigate and compare four variants of the double-pushout approach to graph transformation. Besides the traditional approach with arbitrary matching and injective right-hand morphisms, we consider three variations by employing injective matching and/or arbitrary right-hand morphisms in rules. For each of the three variations, we clarify whether the well-known commutativity theorems are still valid and–where this is not the case–give modified results. In particular, for the most general approach with injective matching and arbitrary right-hand morphisms, we establish sequential and parallel commutativity by appropriately strengthening sequential and parallel independence. We also show that injective matching provides additional expressiveness in two respects, viz. for generating graph languages by grammars without nonterminals and for computing graph functions by convergent graph transformation systems.

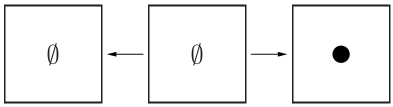
1 Introduction

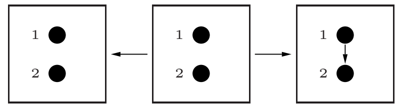
In [EPS73], the first paper on double-pushout graph transformation, matching morphisms are required to be injective. But the vast majority of later papers on the double-pushout approach–including the surveys [Ehr79,CMR+97]–considers arbitrary matching morphisms. Despite this tradition, sometimes it is more natural to require that matching morphisms must be injective. For example, the set of all (directed, unlabelled) loop-free graphs can be generated from the empty

* This research was partly supported by the ESPRIT Working Group APPLIGRAPH. Research of the second author was also supported by the EC TMR Network GETGRATS, through the Universities of Antwerp and Pisa. Part of the work of the third author was done while he was visiting the Vrije Universiteit in Amsterdam.

H. Ehrig et al. (Eds.): Graph Transformation, LNCS 1764, pp. 103–117, 2000.

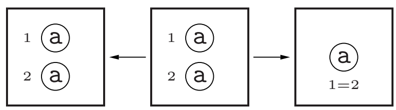
graph by the following two rules if injective matching is assumed:





We will prove that in the traditional double-pushout approach, this graph class cannot be generated without nonterminal labels.

To give a second, non-grammatical example, consider the problem of merging in a graph all nodes labelled with some fixed label a. This is easily achieved–assuming injective matching–by applying as long as possible the following rule:



We will show that without injective matching, there is no finite, convergent graph transformation system–possibly containing identifying rules–solving this problem for arbitrary input graphs. (A graph transformation system is *convergent* if repeated rule applications to an input graph always terminate and yield a unique output graph.)

We consider two variants of the double-pushout approach in which matching morphisms are required to be injective. We denote these approaches by $\mathrm{DPO}^{\mathrm{i/i}}$ and $\mathrm{DPO}^{\mathrm{a/i}}$, where the first component of the exponent indicates whether right-hand morphisms in rules are injective or arbitrary, and where the second component refers to the requirement for matching morphisms. (So our second example belongs to $\mathrm{DPO}^{\mathrm{a/i}}$.)

Besides the traditional approach $\mathrm{DPO}^{\mathrm{i/a}}$, we will also consider $\mathrm{DPO}^{\mathrm{a/a}}$. Obviously, $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{a/i}}$ contain $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$, respectively, as the rules of the latter approaches are included in the former approaches. Moreover, using a quotient construction for rules, we will show that $\mathrm{DPO}^{\mathrm{i/i}}$ and $\mathrm{DPO}^{\mathrm{a/i}}$ can simulate $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$, respectively, in a precise and strong way. Thus the relationships between the approaches can be depicted as follows, where "$\hookrightarrow$" means "is included in" and "$\rightarrow$" means "can be simulated by":

$$\begin{tikzcd}
 & \mathrm{DPO}^{\mathrm{i/a}} \arrow[dl, hook] \arrow[dr] & \\
\mathrm{DPO}^{\mathrm{a/a}} \arrow[dr] & & \mathrm{DPO}^{\mathrm{i/i}} \arrow[dl, hook] \\
 & \mathrm{DPO}^{\mathrm{a/i}} &
\end{tikzcd}$$

The question, then, arises to what extent the theory of $\mathrm{DPO}^{\mathrm{i/a}}$ carries over to the stronger approaches. We answer this question for the classical commutativity theorems, by either establishing their validity or by giving counterexamples and providing modified results.

Lack of space prevents us to give all the proofs of our results. The omitted proofs can be found in the long version of this paper, see [HMP99].

2 Preliminaries

In this section the double-pushout approach to graph transformation is briefly reviewed. For a comprehensive survey, the reader may consult [Ehr79,CMR+97].

A *label alphabet* $\mathcal{C} = \langle \mathcal{C}_V, \mathcal{C}_E \rangle$ is a pair of sets of *node labels* and *edge labels*. A *graph* over $\mathcal{C}$ is a system $G = (\mathrm{V}, \mathrm{E}, \mathrm{s}, \mathrm{t}, \mathrm{l}, \mathrm{m})$ consisting of two finite sets V and E of *nodes* (or *vertices*) and *edges*, two *source* and *target functions* $\mathrm{s}, \mathrm{t} : \mathrm{E} \to \mathrm{V}$, and two *labelling functions* $\mathrm{l} : \mathrm{V} \to \mathcal{C}_V$ and $\mathrm{m} : \mathrm{E} \to \mathcal{C}_E$.

A *graph morphism* $g : G \to H$ between two graphs G and H consists of two functions $g_V : \mathrm{V}_G \to \mathrm{V}_H$ and $g_E : \mathrm{E}_G \to \mathrm{E}_H$ that preserve sources, targets, and labels, that is, $\mathrm{s}_H \circ g_E = g_V \circ \mathrm{s}_G$, $\mathrm{t}_H \circ g_E = g_V \circ \mathrm{t}_G$, $\mathrm{l}_H \circ g_V = \mathrm{l}_G$, and $\mathrm{m}_H \circ g_E = \mathrm{m}_G$. The graphs G and H are the *domain* and *codomain* of g, respectively. A morphism g is *injective* (*surjective*) if g_V and g_E 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 $G \cong H$.

A *rule* $p = \langle L \leftarrow K \to R \rangle$ consists of two graph morphisms with a common domain, where we throughout assume that $K \to L$ is an inclusion. Such a rule is *injective* if $K \to R$ is injective. Given two graphs G and H, G *directly derives* H *through* p, denoted by $G \Rightarrow_p H$, if the diagrams (1) and (2) below are graph pushouts (see [Ehr79] for the definition and construction of graph pushouts).

$$\begin{tikzcd} L \arrow[d] \arrow[dr, phantom, "(1)"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "(2)"] & R \arrow[d] \\ G & D \arrow[l] \arrow[r] & H \end{tikzcd}$$

The notation $G \Rightarrow_{p,g} H$ is used when $g : L \to G$ shall be made explicit.

The *application* of a rule $p = \langle L \leftarrow K \to R \rangle$ to a graph G amounts to the following steps:

(1) Find a graph morphism $g : L \to G$ and check the following *gluing condition*:
Dangling condition. No edge in $G - g(L)$ is incident to a node in $g(L) - g(K)$.
Identification condition. For all distinct items $x, y \in L$, $g(x) = g(y)$ only if $x, y \in K$. (This condition is understood to hold separately for nodes and edges.)
(2) Remove $g(L) - g(K)$ from G, yielding a graph D, a graph morphism $K \to D$ (which is the restriction of g), and the inclusion $D \to G$.
(3) Construct the pushout of $D \leftarrow K \to R$, yielding a graph H and graph morphisms $D \to H \leftarrow R$.

3 Three Variations of the Traditional Approach

A direct derivation $G \Rightarrow_{p,g} H$ is said to be in

- $\mathrm{DPO}^{\mathrm{i/a}}$ if p is injective and g is arbitrary (the "traditional approach"),
- $\mathrm{DPO}^{\mathrm{a/a}}$ if p and g are arbitrary,
- $\mathrm{DPO}^{\mathrm{i/i}}$ if p and g are injective,
- $\mathrm{DPO}^{\mathrm{a/i}}$ if p is arbitrary and g is injective.

Note that in $\mathrm{DPO}^{\mathrm{i/i}}$ and $\mathrm{DPO}^{\mathrm{a/i}}$, step (1) in the application of a rule is simpler than above because the gluing condition reduces to the dangling condition.

We now show that $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$ can be simulated by $\mathrm{DPO}^{\mathrm{i/i}}$ and $\mathrm{DPO}^{\mathrm{a/i}}$, respectively. The idea is to replace in a graph transformation system each rule p by a finite set of rules $\mathrm{Q}(p)$ such that every application of p corresponds to an application of a rule in $\mathrm{Q}(p)$ obeying the injectivity condition.

Definition 1 (Quotient rule). *Given a rule* $p = \langle L \leftarrow K \rightarrow R\rangle$, *a rule* $p' = \langle L' \leftarrow K' \rightarrow R'\rangle$ *is a* quotient rule *of* p *if there is a surjective graph morphism* $K \rightarrow K'$ *such that* L' *and* R' *are the pushout objects of* $L \leftarrow K \rightarrow K'$ *and* $R \leftarrow K \rightarrow K'$, *respectively. The set of quotient rules of* p *is denoted by* $\mathrm{Q}(p)$.

Since the number of non-isomorphic images K' of K is finite, we can without loss of generality assume that $\mathrm{Q}(p)$ is finite. Note also that every quotient rule of an injective rule is injective.

Lemma 1 (Quotient Lemma). *For all graphs* G *and* H, *and all rules* p:

1. $G \Rightarrow_{\mathrm{Q}(p)} H$ *implies* $G \Rightarrow_p H$.
2. $G \Rightarrow_{p,g} H$ *implies* $G \Rightarrow_{p',g'} H$ *for some* $p' \in \mathrm{Q}(p)$ *and some injective* g'.

Proof. 1. Let $G \Rightarrow_{p'} H$ for some $p' = \langle L' \leftarrow K' \rightarrow R'\rangle$ in $\mathrm{Q}(p)$. Then the diagrams (1), (2), (1') and (2') below are graph pushouts. By the composition property for pushouts, the composed diagrams (1)+(1') and (2)+(2') are pushouts as well. Hence $G \Rightarrow_p H$.
2. Let $G \Rightarrow_{p,g} H$ and (e, g') be an epi-mono factorization of g, that is, $e : L \rightarrow L'$ is a surjective morphism and $g' : L' \rightarrow G$ is an injective morphism such that $g = g' \circ e$. Since g satisfies the gluing condition with respect to p, e satisfies the gluing condition with respect to p as well: As e is surjective, it trivially satisfies the dangling condition; since g satisfies the identification condition, e also satisfies this condition. Therefore, a quotient rule $p' = \langle L' \leftarrow K' \rightarrow R'\rangle$ of p can be constructed by a direct derivation $L' \Rightarrow_{p,e} R'$. Moreover, g' satisfies the gluing condition with respect to p': Since g' is injective, g' satisfies the identification condition; since g satisfies the dangling condition with respect to p, g' satisfies the dangling with respect to p'. Therefore, there exists a direct derivation $G \Rightarrow_{p',g'} H'$. Then, by the composition property for pushouts, there is also a direct derivation $G \Rightarrow_{p,g} H'$. Since the result of such a step is unique up to isomorphism, we have $H' \cong H$ and hence $G \Rightarrow_{p',g'} H$.

$$
\begin{tikzcd}
L \arrow[d] \arrow[dr, phantom, "(1)"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "(2)"] & R \arrow[d] \\
L' \arrow[d] \arrow[dr, phantom, "(1')"] & K' \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "(2')"] & R' \arrow[d] \\
G & D \arrow[l] \arrow[r] & H
\end{tikzcd}
\qquad
\begin{tikzcd}
 & L \arrow[dl, dashed] \arrow[dd] & & K \arrow[ll] \arrow[rr] \arrow[dl, dashed] \arrow[dd] & & R \arrow[dl, dashed] \arrow[dd] \\
L' \arrow[dr] & & K' \arrow[ll] \arrow[rr] \arrow[dd, dashed] & & R' \arrow[dd, dashed] & \\
 & G & & D \arrow[ll] \arrow[rr] \arrow[dl, dashed] & & H \arrow[dl, dashed] \\
 & & D' \arrow[ul, dashed] \arrow[rr, dashed] & & H' &
\end{tikzcd}
$$

□

Theorem 1 (Simulation Theorem). *Let $x \in \{\mathrm{a}, \mathrm{i}\}$. Then for all graphs G and H, and every rule p, $G \Rightarrow_p H$ in $\mathrm{DPO}^{x/\mathrm{a}}$ if and only if $G \Rightarrow_{\mathrm{Q}(p)} H$ in $\mathrm{DPO}^{x/\mathrm{i}}$.*

Proof. Immediate consequence of Lemma 1.

4 Parallel Independence

In this section we consider pairs of direct derivations $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ and look for conditions under which there are direct derivations $H \Rightarrow_{p_2} M \Leftarrow_{p_1} \bar{H}$ or $H \Rightarrow_{\mathrm{Q}(p_2)} M \Leftarrow_{\mathrm{Q}(p_1)} \bar{H}$. We fomulate the notions of parallel and strong parallel independence and present three parallel commutativity results.

In the following let $p_i = \langle L_i \leftarrow K_i \to^{r_i} R_i \rangle$, for $i = 1, 2$. (In the diagrams of this section, the morphism r_1 appears to the left of K_1.)

Definition 2 (Parallel independence). *Two direct derivations $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ are* parallelly independent *if there are graph morphisms $L_1 \to D_2$ and $L_2 \to D_1$ such that $L_1 \to D_2 \to G = L_1 \to G$ and $L_2 \to D_1 \to G = L_2 \to G$.*

$$\begin{tikzcd}
R_1 \arrow[d] & K_1 \arrow[l] \arrow[r] \arrow[d] & L_1 \arrow[dr] \arrow[drrr, dashed] \arrow[d, phantom, "="] & & L_2 \arrow[dl] \arrow[dlll, dashed] \arrow[d, phantom, "="] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
H & D_1 \arrow[l] \arrow[rr] & & G & & D_2 \arrow[ll] \arrow[r] & \bar{H}
\end{tikzcd}$$

The notion of parallel independence and its following characterization are well-known, see [EK76] and [ER76], respectively.

Lemma 2 (Characterization of parallel independence). *For $x, y \in \{\mathrm{a}, \mathrm{i}\}$, two direct derivations $H \Leftarrow_{p_1, g_1} G \Rightarrow_{p_2, g_2} \bar{H}$ in $\mathrm{DPO}^{x/y}$ are parallelly independent if and only if the intersection of L_1 and L_2 in G consists of common gluing items, that is, $g_1(L_1) \cap g_2(L_2) \subseteq g_1(K_1) \cap g_2(K_2)$.*

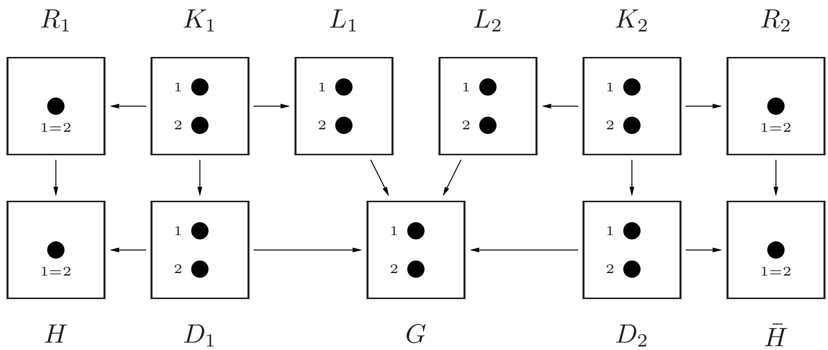
$$\begin{tikzcd}
R_1 \arrow[d] & K_1 \arrow[l] \arrow[r] \arrow[d] & L_1 \arrow[dr, "g_1"'] & & L_2 \arrow[dl, "g_2"] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
H & D_1 \arrow[l] \arrow[rr] & & G & & D_2 \arrow[ll] \arrow[r] & \bar{H}
\end{tikzcd}$$

Theorem 2 (Parallel commutativity I). *In $\mathrm{DPO}^{\mathrm{i/a}}$, $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$, for every pair of parallelly independent direct derivations $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ there are two direct derivations of the form $H \Rightarrow_{p_2} M \Leftarrow_{p_1} \bar{H}$.*

Proof. In [EK76,Ehr79], the statement is proved for direct derivations in $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$. For $\mathrm{DPO}^{\mathrm{i/i}}$, the statement follows by inspecting the proof: If the original derivations are in $\mathrm{DPO}^{\mathrm{i/i}}$, then all morphisms occurring in the proof are injective. Consequently, the direct derivations $H \Rightarrow_{p_2} M \Leftarrow_{p_1} \bar{H}$ are in $\mathrm{DPO}^{\mathrm{i/i}}$. $\square$

The following counterexample demonstrates that the parallel commutativity property does not hold for direct derivations in $\mathrm{DPO}^{\mathrm{a/i}}$.

Example 1. Consider the following direct derivations.



The direct derivations are parallelly independent, but no direct derivations of the form $H \Rightarrow_{p_2} M \Leftarrow_{p_1} \bar{H}$ exist in $\mathrm{DPO}^{\mathrm{a/i}}$. The reason is that the composed morphisms $L_1 \to D_2 \to \bar{H}$ and $L_2 \to D_1 \to H$ are not injective.

This counterexample suggests to strengthen parallel independence as follows.

Definition 3 (Strong parallel independence). *Two direct derivations $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ are* strongly parallelly independent *if there are graph morphisms $L_1 \to D_2$ and $L_2 \to D_1$ such that (a) $L_1 \to D_2 \to G = L_1 \to G$ and $L_2 \to D_1 \to G = L_2 \to G$ and (b) $L_1 \to D_2 \to \bar{H}$ and $L_2 \to D_1 \to H$ are injective.*

Theorem 3 (Parallel commutativity II). *In $\mathrm{DPO}^{\mathrm{a/i}}$, for every pair of strongly parallelly independent direct derivations $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ there are two direct derivations of the form $H \Rightarrow_{p_2} M \Leftarrow_{p_1} \bar{H}$.*

Proof. Let $H \Leftarrow_{p_1,g_1} G \Rightarrow_{p_2,g_2} \bar{H}$ in $\mathrm{DPO}^{\mathrm{a/i}}$ be strongly parallelly independent. Then the two steps are in $\mathrm{DPO}^{\mathrm{a/a}}$ and are parallelly independent. By the proof of Theorem 2, there are direct derivations $H \Rightarrow_{p_2,g_2'} M \Leftarrow_{p_1,g_1'} \bar{H}$ in $\mathrm{DPO}^{\mathrm{a/a}}$ such that $g_2' = L_2 \to D_1 \to H$ and $g_1' = L_1 \to D_2 \to \bar{H}$. Since both morphisms are injective by strong parallel independence, the derivations are in $\mathrm{DPO}^{\mathrm{a/i}}$. □

Note that by Theorem 2 and the fact that strong parallel independence implies parallel independence, Theorem 3 holds for $\mathrm{DPO}^{\mathrm{i/a}}$, $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$ as well.

Theorem 4 (Parallel commutativity III). *In $\mathrm{DPO}^{\mathrm{a/i}}$, for every pair of parallelly independent direct derivations $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ there are two direct derivations of the form $H \Rightarrow_{\mathrm{Q}(p_2)} M \Leftarrow_{\mathrm{Q}(p_1)} \bar{H}$.*

Proof. Let $H \Leftarrow_{p_1} G \Rightarrow_{p_2} \bar{H}$ in $\mathrm{DPO}^{\mathrm{a/i}}$ be parallelly independent. Then the derivations are in $\mathrm{DPO}^{\mathrm{a/a}}$ and, by Theorem 2, there are direct derivations $H \Rightarrow_{p_2} M \Leftarrow_{p_1} \bar{H}$ in $\mathrm{DPO}^{\mathrm{a/a}}$. Hence, by the Simulation Theorem, there are direct derivations $H \Rightarrow_{\mathrm{Q}(p_2)} M \Leftarrow_{\mathrm{Q}(p_1)} \bar{H}$ in $\mathrm{DPO}^{\mathrm{a/i}}$. $\square$

By Theorem 2 and the fact that every rule p is contained in $\mathrm{Q}(p)$, Theorem 4 holds for $\mathrm{DPO}^{\mathrm{i/a}}$, $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$ as well.

5 Sequential Independence

We now switch from parallel independence to sequential independence, looking for conditions under which two consecutive direct derivations can be interchanged.

Definition 4 (Sequential independence). *Two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ are* sequentially independent *if there are morphisms $R_1 \to D_2$ and $L_2 \to D_1$ such that $R_1 \to D_2 \to H = R_1 \to H$ and $L_2 \to D_1 \to H = L_2 \to H$.*

$$\begin{tikzcd}
L_1 \arrow[d] & K_1 \arrow[l] \arrow[r] \arrow[d] & R_1 \arrow[dr] \arrow[drrr, dashed] & & L_2 \arrow[dl] \arrow[dlll, dashed] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
G & D_1 \arrow[l] \arrow[rr] & \arrow[r, phantom, "="] & H & \arrow[l, phantom, "="] & D_2 \arrow[ll] \arrow[r] & M
\end{tikzcd}$$

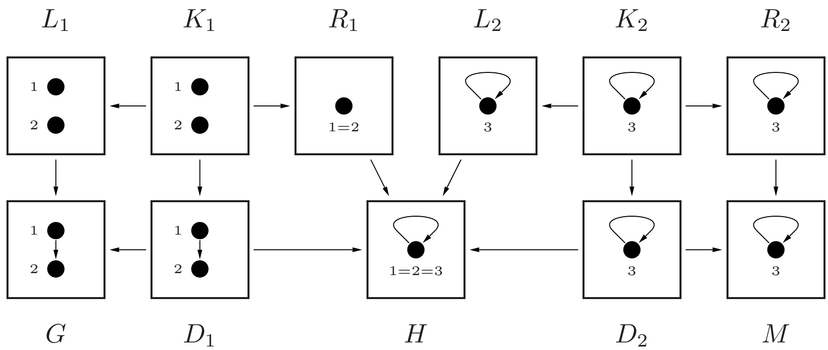
The following characterization of sequential independence can be proved analogously to Lemma 2, see [ER76].

Lemma 3 (Characterization of sequential independence). *For $y \in \{\mathrm{a}, \mathrm{i}\}$, two direct derivations $G \Rightarrow_{p_1,g_1} H \Rightarrow_{p_2,g_2} M$ in $\mathrm{DPO}^{\mathrm{i}/y}$ are sequentially independent if and only if the intersection of R_1 and L_2 in H consists of common gluing items, that is, $h_1(R_1) \cap g_2(L_2) \subseteq h_1(r_1(K_1)) \cap g_2(K_2)$.*

$$\begin{tikzcd}
L_1 \arrow[d] & K_1 \arrow[l] \arrow[r, "r_1"] \arrow[d] & R_1 \arrow[dr, "h_1"'] & & L_2 \arrow[dl, "g_2"] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
G & D_1 \arrow[l] \arrow[rr] & & H & & D_2 \arrow[ll] \arrow[r] & M
\end{tikzcd}$$

This characterization may break down, however, in the presence of non-injective rules. So Lemma 3 does not hold in $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{a/i}}$, as is demonstrated by the next example.

Example 2. Consider the following direct derivations.

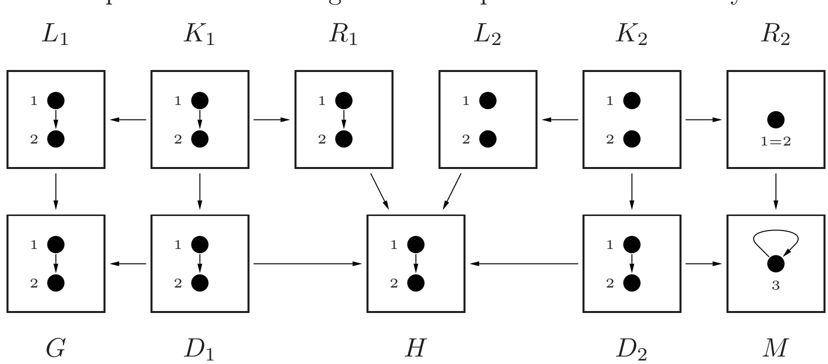


The intersection of R_1 and L_2 in H consists of common gluing items. But there does not exist a graph morphism $L_2 \to D_1$ (see also [Mül97], Example 2.5).

Theorem 5 (Sequential commutativity I). *In* $\mathrm{DPO}^{\mathrm{i/a}}$, $\mathrm{DPO}^{\mathrm{a/a}}$, $\mathrm{DPO}^{\mathrm{i/i}}$, *for every pair of sequentially independent direct derivations* $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ *there are two sequentially independent direct derivations* $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$.

Proof. In [EK76], the statement is proved for direct derivations in $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$. For $\mathrm{DPO}^{\mathrm{i/i}}$, the statement follows by inspecting the proof: If the original derivations are in $\mathrm{DPO}^{\mathrm{i/i}}$, then all morphisms occurring in the proof become injective. Consequently, the direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ are in $\mathrm{DPO}^{\mathrm{i/i}}$. (A self-contained proof of Theorem 5 can be found in [HMP99].) □

Example 3. The following direct derivations demonstrate that in $\mathrm{DPO}^{\mathrm{a/i}}$, sequential independence does not guarantee sequential commutativity:



The two steps are sequentially independent. Moreover, there exists a direct derivation of the form $G \Rightarrow_{p_2} \bar{H}$ in $\mathrm{DPO}^{\mathrm{a/i}}$. But there is no step $\bar{H} \Rightarrow_{p_1} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$. The reason is that the composed morphism $R_1 \to D_2 \to M$ is not injective (see also [Mül97], Example 2.1).

We now strengthen sequential independence by requiring that $R_1 \to D_2 \to M$ is injective. We need not require $L_2 \to D_1 \to G$ to be injective, though, because this follows from the injectivity of $L_2 \to H$ and $K_1 \to L_1$.

Definition 5 (Strong sequential independence). *Two direct derivations* $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ *are* strongly sequentially independent *if there are graph morphisms* $R_1 \to D_2$ *and* $L_2 \to D_1$ *such that (a)* $R_1 \to D_2 \to H = R_1 \to H$ *and* $L_2 \to D_1 \to H = L_2 \to H$ *and (b)* $R_1 \to D_2 \to M$ *is injective.*

Theorem 6 (Sequential commutativity II). *In* $\mathrm{DPO}^{\mathrm{a/i}}$*, for every pair of strongly sequentially independent direct derivations* $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ *there are two sequentially independent direct derivations of the form* $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$*.*

Since strong sequential independence implies sequential independence, Theorem 6 also holds for $\mathrm{DPO}^{\mathrm{i/a}}$, $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$. Moreover, it can be shown that the direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ are even strongly sequentially independent in $\mathrm{DPO}^{\mathrm{a/i}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$, but this need not hold in $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$.

Next we combine Theorem 5 with the Simulation Theorem.

Theorem 7 (Sequential commutativity III). *In* $\mathrm{DPO}^{\mathrm{a/i}}$*, for every pair of sequentially independent direct derivations* $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ *there are two direct derivations of the form* $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{\mathrm{Q}(p_1)} M$*.*

Proof. Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$ be sequentially independent. Then the direct derivations are in $\mathrm{DPO}^{\mathrm{a/a}}$ and, by Theorem 5, there are sequentially independent direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ in $\mathrm{DPO}^{\mathrm{a/a}}$. Since the original derivations are in $\mathrm{DPO}^{\mathrm{a/i}}$, the morphism $L_2 \to G$ is injective. By the Simulation Theorem, there is a direct derivation $\bar{H} \Rightarrow_{\mathrm{Q}(p_1)} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$. Therefore, there are two direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{\mathrm{Q}(p_1)} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$. □

To see that the direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{\mathrm{Q}(p_1)} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$ need not be sequentially independent, consider the sequentially independent direct derivations of Example 3. There the resulting steps $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{\mathrm{Q}(p_1)} M$ are not sequentially independent.

By Theorem 5 and the fact that every rule p is contained in $\mathrm{Q}(p)$, Theorem 7 holds for $\mathrm{DPO}^{\mathrm{i/a}}$, $\mathrm{DPO}^{\mathrm{a/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$ as well.

6 Expressiveness

In this section we show that injective matching provides additional expressiveness in two respects. First we study the generative power of graph grammars without nonterminal labels and show that these grammars can generate in $\mathrm{DPO}^{x/\mathrm{i}}$ more languages than in $\mathrm{DPO}^{x/\mathrm{a}}$, for $x \in \{\mathrm{a}, \mathrm{i}\}$. Then we consider the computation of functions on graphs by convergent graph transformation systems and prove that in $\mathrm{DPO}^{\mathrm{a/i}}$ more functions can be computed than in $\mathrm{DPO}^{\mathrm{a/a}}$.

Given two graphs G, H and a set of rules $\mathcal{R}$, we write $G \Rightarrow_{\mathcal{R}} H$ if there is a rule p in $\mathcal{R}$ such that $G \Rightarrow_p H$. A *derivation* from G to H over $\mathcal{R}$ is a sequence of the form $G = G_0 \Rightarrow_{\mathcal{R}} G_1 \Rightarrow_{\mathcal{R}} \ldots \Rightarrow_{\mathcal{R}} G_n \cong H$, which may be denoted by $G \Rightarrow_{\mathcal{R}}^* H$.

6.1 Generative Power

We study the expressiveness of graph grammars in $\mathrm{DPO}^{x/y}$, for $x, y \in \{\mathrm{a}, \mathrm{i}\}$. It turns out that all four approaches have the same generative power for unrestricted grammars, but $\mathrm{DPO}^{x/\mathrm{i}}$ is more powerful than $\mathrm{DPO}^{x/\mathrm{a}}$ if we consider grammars without nonterminal labels. In particular, we prove that there is a grammar without nonterminal labels in $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$ generating a language that cannot be generated by any grammar without nonterminal labels in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$.

Let $x, y \in \{\mathrm{a}, \mathrm{i}\}$. A *graph grammar in* $\mathrm{DPO}^{x/y}$ is a system $\mathcal{G} = \langle \mathcal{C}, \mathcal{R}, S, \mathcal{T} \rangle$, where $\mathcal{C}$ is a finite label alphabet, $\mathcal{T}$ is an alphabet of *terminal* labels with $\mathcal{T}_V \subseteq \mathcal{C}_V$ and $\mathcal{T}_E \subseteq \mathcal{C}_E$, S is a graph over $\mathcal{C}$ called the *start graph*, and $\mathcal{R}$ is a finite set of rules over $\mathcal{C}$ such that if $x = \mathrm{i}$, all rules in $\mathcal{R}$ are injective. The *graph language generated by* $\mathcal{G}$ is the set $\mathrm{L}(\mathcal{G})$ consisting of all graphs G over $\mathcal{T}$ such that there is a derivation $S \Rightarrow^*_{\mathcal{R}} G$ in $\mathrm{DPO}^{x/y}$. We denote by $\mathcal{L}^{x/y}$ the class of all graph languages generated by graph grammars in $\mathrm{DPO}^{x/y}$.

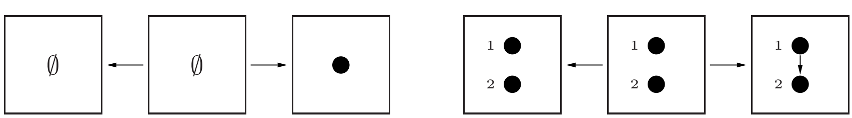
Theorem 8. $\mathcal{L}^{\mathrm{i}/\mathrm{a}} = \mathcal{L}^{\mathrm{i}/\mathrm{i}} = \mathcal{L}^{\mathrm{a}/\mathrm{a}} = \mathcal{L}^{\mathrm{a}/\mathrm{i}}$.

Next we put Theorem 8 into perspective by showing that injective matching provides more generative power if we restrict ourselves to grammars without nonterminal labels. To this end we denote by $\mathcal{L}^{x/y}_{\mathrm{T}}$ the class of all graph languages generated by a grammar $\langle \mathcal{C}, \mathcal{R}, S, \mathcal{T} \rangle$ in $\mathrm{DPO}^{x/y}$ with $\mathcal{T} = \mathcal{C}$. Note that for such a grammar, every graph derivable from S belongs to the generated language.

Theorem 9. *For* $x \in \{\mathrm{a}, \mathrm{i}\}$, $\mathcal{L}^{x/\mathrm{a}}_{\mathrm{T}} \subset \mathcal{L}^{x/\mathrm{i}}_{\mathrm{T}}$.

Proof. Let $\mathcal{G} = \langle \mathcal{C}, \mathcal{R}, S, \mathcal{T} \rangle$ be a grammar without nonterminal labels in $\mathrm{DPO}^{x/\mathrm{a}}$ and $\mathrm{Q}(\mathcal{R}) = \{\mathrm{Q}(p) \mid p \in \mathcal{R}\}$. Then $\mathrm{Q}(\mathcal{G}) = \langle \mathcal{C}, \mathrm{Q}(\mathcal{R}), S, \mathcal{T} \rangle$ is a grammar without nonterminal labels in $\mathrm{DPO}^{x/\mathrm{i}}$ such that, by the Simulation Theorem, $\mathrm{L}(\mathcal{G}) = \mathrm{L}(\mathrm{Q}(\mathcal{G}))$. Hence $\mathcal{L}^{x/\mathrm{a}}_{\mathrm{T}} \subseteq \mathcal{L}^{x/\mathrm{i}}_{\mathrm{T}}$.

To show that the inclusion is strict, let $\mathcal{G} = \langle \mathcal{C}, \mathcal{R}, S, \mathcal{T} \rangle$ be the grammar in $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$ where S is the empty graph, $\mathcal{C}_{\mathrm{V}}$ and $\mathcal{C}_{\mathrm{E}}$ are singletons, $\mathcal{T} = \mathcal{C}$, and $\mathcal{R}$ consists of the following two rules (already shown in the introduction):



We will show that no grammar in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$ without nonterminal labels can generate $\mathrm{L}(\mathcal{G})$, the set of all loop-free graphs over $\mathcal{C}$. To this end, suppose the contrary and consider a grammar $\mathcal{G}' = \langle \mathcal{C}, \mathcal{R}', S', \mathcal{T} \rangle$ in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$ such that $\mathrm{L}(\mathcal{G}') = \mathrm{L}(\mathcal{G})$.

First we show that rules that identify nodes cannot occur in derivations of graphs in $\mathrm{L}(\mathcal{G}')$. Let $p = \langle L \leftarrow K \rightarrow^r R \rangle$ be a rule in $\mathcal{R}'$ such that $r_{\mathrm{V}}(v) = r_{\mathrm{V}}(v')$ for two distinct nodes v and v' in K. Then L cannot be loop-free as otherwise $L^{\oplus}$, obtained from L by adding an edge between v and v', would also be loop-free and hence belong to $\mathrm{L}(\mathcal{G}')$. The latter is impossible since $L^{\oplus} \Rightarrow_p R^{\oplus} \notin \mathrm{L}(\mathcal{G}')$, where $R^{\oplus}$ is obtained from R by attaching a loop to $r_{\mathrm{V}}(v)$. Thus each rule in $\mathcal{R}'$ that identifies nodes must contain a loop in its left-hand side, and hence it cannot occur in any derivation of a graph in $\mathrm{L}(\mathcal{G}')$.

The idea, now, is to show that in every derivation of a complete graph of sufficient size, a rule of $\mathcal{R}'$ must be applied that creates an edge between two existing nodes. This fact contradicts the absence of loops in $\mathrm{L}(\mathcal{G}')$, as will be easy to show.

Let k be the maximal number of nodes occurring in S' or in a right-hand side of $\mathcal{R}'$. Consider a loop-free graph G with $k+1$ nodes such that there is an edge between each two distinct nodes. Let $S' \cong G_0 \Rightarrow_{\mathcal{R}'} G_1 \Rightarrow_{\mathcal{R}'} \ldots \Rightarrow_{\mathcal{R}'} G_n = G$ be a derivation generating G, and $i \in \{0, \ldots, n-1\}$ the largest index such that $G_i \Rightarrow_{\mathcal{R}'} G_{i+1}$ creates a node v that is not removed in the rest derivation $G_{i+1} \Rightarrow^*_{\mathcal{R}'} G_n$. Then $\mathrm{V}_G = \mathrm{V}_{G_n}$ is contained in $\mathrm{V}_{G_{i+1}}$ up to isomorphism. W.l.o.g. we assume $\mathrm{V}_G \subseteq \mathrm{V}_{G_{i+1}}$. Since V_G contains more nodes than the right-hand side of the rule applied in $G_i \Rightarrow_{\mathcal{R}'} G_{i+1}$, there must exist a node v' in $\mathrm{V}_G \subseteq \mathrm{V}_{G_{i+1}}$ that is not in the image of the right-hand side. Thus, because v is created in $G_i \Rightarrow_{\mathcal{R}'} G_{i+1}$, there is no edge between v and v' in G_{i+1}. As there is an edge between v and v' in G, and $G_{i+1} \Rightarrow^*_{\mathcal{R}'} G$ does not identify nodes, there is a step $G_j \Rightarrow_{\mathcal{R}'} G_{j+1}$ with $j \geq i+1$ that creates an edge between v and v' while there is no such edge in G_j. Let $p = \langle L \leftarrow K \rightarrow R \rangle$ be the rule applied in $G_j \Rightarrow_{\mathcal{R}'} G_{j+1}$. Then there are two distinct nodes v_1 and v_2 in K such that there is an edge between (the images of) v_1 and v_2 in R but not in L.

Next observe that L is loop-free because G_j is. So $\tilde{L}$, obtained from L by identifying v_1 and v_2, is loop-free as well. On the other hand, there is a step $\tilde{L} \Rightarrow_{p,g} \tilde{R}$ where $g\colon L \rightarrow \tilde{L}$ is the surjective morphism associated with the construction of $\tilde{L}$. But then $\tilde{R}$ contains a loop, contradicting the fact that $\tilde{L}$ belongs to $\mathrm{L}(\mathcal{G}')$. □

6.2 DPO-Computable Functions

Graph transformation systems that transform (or "reduce") every graph into a unique irreducible graph provide a natural model for computing functions on graphs. Since the graphs resulting from double-pushout derivations are unique only up to isomorphism, we consider derivations and functions on isomorphism classes of graphs.

An *abstract graph* over a label alphabet $\mathcal{C}$ is an isomorphism class of graphs over $\mathcal{C}$. We write $[G]$ for the isomorphism class of a graph G, and $\mathcal{A}_{\mathcal{C}}$ for the set of all abstract graphs over $\mathcal{C}$. A *graph transformation system in* $\mathrm{DPO}^{x/y}$, for $x, y \in \{\mathrm{a}, \mathrm{i}\}$, is a pair $\langle \mathcal{C}, \mathcal{R} \rangle$ where $\mathcal{C}$ is a label alphabet and $\mathcal{R}$ a set of rules with graphs over $\mathcal{C}$ such that if $x = \mathrm{i}$, then all rules are injective. Such a system is *finite* if $\mathcal{C}_{\mathrm{V}}$, $\mathcal{C}_{\mathrm{E}}$ and $\mathcal{R}$ are finite sets. We will often identify $\langle \mathcal{C}, \mathcal{R} \rangle$ with $\mathcal{R}$, leaving $\mathcal{C}$ implicit. The relation $\Rightarrow_{\mathcal{R}}$ is lifted to $\mathcal{A}_{\mathcal{C}}$ by

$$[G] \Rightarrow_{\mathcal{R}} [H] \text{ if } G \Rightarrow_{\mathcal{R}} H.$$

This yields a well-defined relation since for all graphs G, G', H, H' over $\mathcal{C}$, $G' \cong G \Rightarrow_{\mathcal{R}} H \cong H'$ implies $G' \Rightarrow_{\mathcal{R}} H'$.

A graph transformation system $\mathcal{R}$ is *terminating* if there is no infinite sequence $G_1 \Rightarrow_{\mathcal{R}} G_2 \Rightarrow_{\mathcal{R}} \ldots$ of graphs in $\mathcal{A}_{\mathcal{C}}$. Let $\Rightarrow^*_{\mathcal{R}}$ be the transitive-reflexive closure of $\Rightarrow_{\mathcal{R}}$. Then $\mathcal{R}$ is *confluent* (or has the *Church-Rosser property*) if for all $G, H_1, H_2 \in \mathcal{A}_{\mathcal{C}}$, $H_1 \Leftarrow^*_{\mathcal{R}} G \Rightarrow^*_{\mathcal{R}} H_2$ implies that there is some $H \in \mathcal{A}_{\mathcal{C}}$ such that $H_1 \Rightarrow^*_{\mathcal{R}} H \Leftarrow^*_{\mathcal{R}} H_2$. If $\mathcal{R}$ is both terminating and confluent, then it is *convergent*. An abstract graph $G \in \mathcal{A}_{\mathcal{C}}$ is a *normal form* (with respect to $\mathcal{R}$) if there is no $H \in \mathcal{A}_{\mathcal{C}}$ such that $G \Rightarrow_{\mathcal{R}} H$. If $\mathcal{R}$ is convergent, then for every abstract graph G over $\mathcal{C}$ there is a unique normal form N such that $G \Rightarrow^*_{\mathcal{R}} N$. In this case we denote by $\mathrm{N}_{\mathcal{R}}$ the function on $\mathcal{A}_{\mathcal{C}}$ that sends every abstract graph to its normal form.

Definition 6 (DPO-computable function). *Let $x, y \in \{\mathrm{a}, \mathrm{i}\}$. A function $f\colon \mathcal{A}_{\mathcal{C}} \to \mathcal{A}_{\mathcal{C}}$ is* computable in $\mathrm{DPO}^{x/y}$, *or* $\mathrm{DPO}^{x/y}$-computable *for short, if there exists a finite, convergent graph transformation system $\langle \mathcal{C}, \mathcal{R} \rangle$ in $\mathrm{DPO}^{x/y}$ such that $\mathrm{N}_{\mathcal{R}} = f$.*

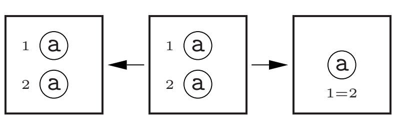
Theorem 10. *Each function that is computable in $\mathrm{DPO}^{x/\mathrm{a}}$ is also computable in $\mathrm{DPO}^{x/\mathrm{i}}$, for $x \in \{\mathrm{a}, \mathrm{i}\}$.*

Proof. Let $\langle \mathcal{C}, \mathcal{R} \rangle$ be a finite, convergent graph transformation system in $\mathrm{DPO}^{x/\mathrm{a}}$. Then $\langle \mathcal{C}, \mathrm{Q}(\mathcal{R}) \rangle$ is a finite and convergent system in $\mathrm{DPO}^{x/\mathrm{i}}$, as can easily be checked by means of the Simulation Theorem. Moreover, by the same result, every abstract graph has the same normal form with respect to $\mathcal{R}$ and $\mathrm{Q}(\mathcal{R})$, respectively. Thus $\mathrm{N}_{\mathcal{R}} = \mathrm{N}_{\mathrm{Q}(\mathcal{R})}$, which implies the proposition. □

So the double-pushout approach with injective matching is at least as powerful for computing functions as the traditional approach with arbitrary matching. The next result shows that–at least if identifying rules are present–injective matching indeed provides additional power.

Theorem 11. *There exists a function that is computable in $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$ but not in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$.*

Proof. Let $\mathcal{C}$ be a label alphabet and $\mathtt{a} \in \mathcal{C}_{\mathrm{V}}$. Consider the function $f\colon \mathcal{A}_{\mathcal{C}} \to \mathcal{A}_{\mathcal{C}}$ sending each abstract graph $[G]$ to $[G']$, where G' is obtained from G by merging all nodes labelled with $\mathtt{a}$. Let $\mathcal{R}$ be the system in $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$ consisting of the following single rule:



This system is terminating as each direct derivation reduces the number of nodes by one. Moreover, $\mathcal{R}$ has the following *subcommutativity property*: Whenever $H_1 \Leftarrow_{\mathcal{R}} G \Rightarrow_{\mathcal{R}} H_2$ for some $G, H_1, H_2 \in \mathcal{A}_{\mathcal{C}}$, then $H_1 = H_2$ or there is some $H \in \mathcal{A}_{\mathcal{C}}$ such that $H_1 \Rightarrow_{\mathcal{R}} H \Leftarrow_{\mathcal{R}} H_2$. This is easy to verify with the help of Theorem 3 (parallel commutativity in $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$). It follows that $\mathcal{R}$ is confluent, for this is a well-known consequence of subcommutativity [Hue80]. Then it is evident that $\mathrm{N}_{\mathcal{R}} = f$, that is, f is computable in $\mathrm{DPO}^{\mathrm{a/i}}$.

To show that f cannot be computed in $\mathrm{DPO}^{\mathrm{a/a}}$, suppose the contrary. Let $\mathcal{R}'$ be a finite, convergent graph transformation system in $\mathrm{DPO}^{\mathrm{a/a}}$ such that $\mathrm{N}_{\mathcal{R}'} = f$. Since $\mathcal{R}'$ is finite, there is a maximal number k of edges occurring in a left-hand side of a rule in $\mathcal{R}'$. Define G to be a graph consisting of two nodes, both labelled with $\mathtt{a}$, and $k+1$ edges between these nodes. (The labels and directions of these edges do not matter.) Now consider a derivation $[G] \Rightarrow^*_{\mathcal{R}'} f([G])$. This derivation contains at least one step since $[G] \neq f([G])$. Let $[G] \Rightarrow_{\mathcal{R}'} [H]$ be the first step of the derivation, with an underlying step $G \Rightarrow_{p,g} H$ on graphs. By the dangling condition, p cannot remove one of the two nodes in G. For, both nodes are incident to $k+1$ edges while p can remove at most k edges. Next consider some graph G' in $f([G])$ (consisting of a single node labelled with $\mathtt{a}$ and $k+1$ loops) and a surjective graph morphism $h\colon G \to G'$. Since p does not remove nodes and h is injective on edges, the composed morphism $h \circ g$ satisfies the gluing condition. So there is a step $G' \Rightarrow_{p,\, h \circ g} X$ for some X. But G' is in $f([G]) = \mathrm{N}_{\mathcal{R}'}([G])$ and hence is a normal form. Thus our assumption that f can be computed in $\mathrm{DPO}^{\mathrm{a/a}}$ has led to a contradiction. $\square$

7 Conclusion

We have shown that injective matching makes double-pushout graph transformation more expressive. This applies to both the generative power of grammars without nonterminals and the computability of functions by convergent graph transformation systems.

The classical independence results of the double-pushout approach have been reconsidered for three variations of the traditional approach, and have been adapted where necessary. These results can be summarized as follows, where "yes" indicates a positive result and "no" means that there exists a counterexample:

	$\mathrm{DPO}^{\mathrm{i/a}}$	$\mathrm{DPO}^{\mathrm{a/a}}$	$\mathrm{DPO}^{\mathrm{i/i}}$	$\mathrm{DPO}^{\mathrm{a/i}}$
char. of parallel independence	yes	yes	yes	yes
char. of seqential independence	yes	no	yes	no
parallel commutativity I	yes	yes	yes	no
parallel commutativity II & III	yes	yes	yes	yes
sequential commutativity I	yes	yes	yes	no
sequential commutativity II & III	yes	yes	yes	yes

Corresponding results on parallelism and concurrency can be found in the long version [HMP99]. A topic for future work is to address the classical results on canonical derivations and amalgamation in the double-pushout approach.

One may also consider injective matching in the *single-pushout approach* [Löw93]. In [HHT96], parallel and sequential independence are studied for single-pushout derivations with negative application conditions. These include the dangling and the injectivity condition for matching morphisms. However, the definition of sequential independence in [HHT96] requires more information on the given direct derivations as our definition of strong sequential independence.

A further topic is to consider *high-level replacement systems*, that is, double-pushout derivations in arbitrary categories. We just mention that our independence results do not follow from the ones in [EHKP91], where non-injective rules and injective matching morphisms are not considered.

References

CMR+97. Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, Reiko Heckel, Michael Löwe. Algebraic approaches to graph transformation. Part I: Basic concepts and double pushout approach. In G. Rozenberg, ed., Handbook of Graph Grammars and Computing by Graph Transformation, volume 1: Foundations, chapter 3, 163–245. World Scientific, 1997. 103, 105

Ehr79. Hartmut Ehrig. Introduction to the algebraic theory of graph grammars. In Proc. Graph-Grammars and Their Application to Computer Science and Biology, Lecture Notes in Computer Science 73, 1–69, 1979. 103, 105, 107

EHKP91. Hartmut Ehrig, Annegret Habel, Hans-Jörg Kreowski, Francesco Parisi-Presicce. Parallelism and concurrency in high level replacement systems. Mathematical Structures in Computer Science 1, 361–404, 1991. 116

EK76. Hartmut Ehrig, Hans-Jörg Kreowski. Parallelism of manipulations in multidimensional information structures. In Proc. Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 45, 284–293, 1976. 107, 110

EK79. Hartmut Ehrig, Hans-Jörg Kreowski. Pushout-properties: An analysis of gluing constructions for graphs. Mathematische Nachrichten 91, 135–149, 1979.

ER76. Hartmut Ehrig, Barry K. Rosen. Commutativity of independent transformations on complex objects. IBM Research Report RC 6251, Yorktown Heights, 1976. 107, 109

EPS73. Hartmut Ehrig, Michael Pfender, Hans Jürgen Schneider. Graph grammars: An algebraic approach. In Proc. IEEE Conf. on Automata and Switching Theory, 167–180, 1973. 103

HHT96. Annegret Habel, Reiko Heckel, Gabriele Taentzer. Graph grammars with negative application conditions. Fundamenta Informaticae 26, 287–313, 1996. 115, 116

HMP99. Annegret Habel, Jürgen Müller, Detlef Plump. Double-pushout graph transformation revisited. Technical report 7/99, Fachbereich Informatik, Universität Oldenburg, 1999. 104, 110, 115

Hue80. Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. Journal of the ACM 27(4), 797–821, 1980. 115

Löw93. Michael Löwe. Algebraic approach to single-pushout graph transformation. Theoretical Computer Science 109, 181–224, 1993. 115

Mül97. Jürgen Müller. A non-categorical characterization of sequential independence for algebraic graph rewriting and some applications. Technical report 97-18, Technische Universität Berlin, Fachbereich Informatik, 1997. 110