Math. Struct. in Comp. Science (2001), *vol.* 11, *pp.* 637–688. © 2001 Cambridge University Press
DOI: 10.17/S0960129501003425 Printed in the United Kingdom

Double-pushout graph transformation revisited[†]

ANNEGRET HABEL[‡], JÜRGEN MÜLLER[§] and DETLEF PLUMP[¶]

[‡]*Universität Oldenburg, Fachbereich Informatik, 26111 Oldenburg, Germany*

[§]*28 Chichester Road, Seaford, East Sussex BN25 2DL, United Kingdom*

[¶]*The University of York, Department of Computer Science, York YO10 5DD, United Kingdom*

Received 20 August 1999; revised 31 August 2000

In this paper we investigate and compare four variants of the double-pushout approach to graph transformation. As well as 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. We show that injective matching provides additional expressiveness in two respects: for generating graph languages by grammars without non-terminals and for computing graph functions by convergent graph transformation systems. Then we clarify for each of the three variations whether the well-known commutativity, parallelism and concurrency 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.

1. Introduction

The double-pushout approach to graph transformation was introduced in 1973 by the paper Ehrig *et al.* (1973). Since then the approach has been studied extensively and has also been applied to several areas of computer science: see, for example, the three volumes of the *Handbook on Graph Grammars and Computing by Graph Transformation* (Rozenberg 1997; Ehrig *et al.* 1999a; Ehrig *et al.* 1999b). In the original paper (Ehrig *et al.* 1973), the left-hand sides of transformation rules are matched by injective graph morphisms, but the vast majority of later papers on the double-pushout approach – including the surveys Ehrig (1979) and Corradini *et al.* (1997) – consider arbitrary, possibly non-injective matching morphisms. In this paper we argue that despite this tradition, injective matching

[†] This research was partly supported by the ESPRIT Working Group APPLIGRAPH.
This paper is an extension of Habel *et al.* (2000).
[‡] J. Müller's contribution was also supported by the EC TMR Network GETGRATS, through the Universities of Antwerp and Pisa.
[§] D. Plump's contribution was done while he was with Universität Bremen; part of his work was done during a visit at the Vrije Universiteit in Amsterdam.

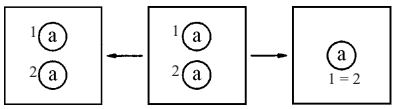
https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

makes the approach more expressive. For example, the set of all (directed, unlabelled) loop-free graphs can be generated from the empty graph by the following two rules if injective matching is assumed:



We will prove that in the traditional double-pushout approach with arbitrary matching, this graph class cannot be generated without using non-terminal labels.

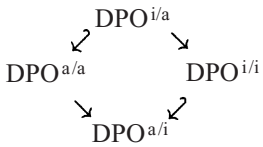
To give a second, non-grammatical example, consider the problem of merging in a graph all nodes labelled with some fixed label 'a'. This is easily achieved – assuming injective matching – by applying for as long as possible the following rule:



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 $\mathrm{DPO}^{i/i}$ and $\mathrm{DPO}^{a/i}$, 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 $\mathrm{DPO}^{a/i}$.)

Besides the traditional approach $\mathrm{DPO}^{i/a}$, we will also consider $\mathrm{DPO}^{a/a}$. Obviously, $\mathrm{DPO}^{a/a}$ and $\mathrm{DPO}^{a/i}$ contain $\mathrm{DPO}^{i/a}$ and $\mathrm{DPO}^{i/i}$, 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 $\mathrm{DPO}^{i/i}$ and $\mathrm{DPO}^{a/i}$ can simulate $\mathrm{DPO}^{i/a}$ and $\mathrm{DPO}^{a/a}$, 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':



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

The paper is organized as follows. Section 2 provides the basic notions of the double-pushout approach. In Section 3 we present three variations of the traditional approach and establish a simulation theorem saying that graph transformation with arbitrary matching can be simulated by graph transformation with injective matching. In Section

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

4 we show that injective matching provides additional expressiveness both with respect to the generative power of certain graph grammars and the computability of functions by convergent graph transformation systems. Motivated by this observation, in Sections 5, 6, 7 and 8 we present a systematic study of the three variations of the traditional approach and establish results on parallel and sequential commutativity, parallelism and concurrency. This is done by analogy with the well-known results for the traditional approach. In Section 9 we relate the results for parallelism and concurrency, and show that the parallelism theorems can be seen as special cases of the concurrency theorems. In Section 10 we conclude with a brief summary and with some topics for future work. Finally, the Appendix summarizes some facts about pushouts and pullbacks.

2. Preliminaries

This section briefly reviews the basic notions of the double-pushout approach to graph transformation. We consider here $\mathrm{DPO}^{\mathrm{a/a}}$ – arbitrary morphisms both in the right-hand sides of rules and for matching – to obtain three other variants in the next section by requiring injectivity for one or both of the relevant morphisms. For a comprehensive survey of the 'traditional approach' $\mathrm{DPO}^{\mathrm{i/a}}$, the reader may consult Ehrig (1979) and Corradini *et al.* (1997)[†].

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

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 $G \cong H$. An injective morphism g is an *inclusion* if $g_V(v) = v$ and $g_E(e) = e$ for all $v \in V_G$ and $e \in E_G$. The *composition* of $g : G \to H$ with a morphism $h : H \to M$ is the morphism $h \circ g : G \to M$ consisting of the composed functions $h_V \circ g_V$ and $h_E \circ g_E$. The composition is also written $G \to H \to M$ if g and h are clear from the context.

A *rule* (or *production*) $p = \langle L \leftarrow K \rightarrow R \rangle$ consists of two graph morphisms with a common domain. L is the *left-hand side*, R the *right-hand side*, and K the *interface* of p. Such a rule is *injective* if $K \to R$ is injective. We will assume throughout that $K \to L$ is an inclusion. Allowing non-injective morphisms $K \to L$ would have the undesirable effect that applying a rule with a given matching morphism becomes non-deterministic in that the graphs D and H in Figure 1 below need not be unique up to isomorphism (see Rosen (1975)); hence we refrain from allowing such rules.

[†] In the mid 1970's, several papers of the Berlin school considered both $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$. But $\mathrm{DPO}^{\mathrm{i/a}}$ has been the predominant approach in the literature since at least 1979, which is why we call this approach 'traditional'.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

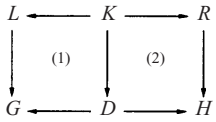


Fig. 1. A direct derivation

Given two graphs G and H, G *directly derives* H *through* p, denoted by $G \Rightarrow_p H$, if the diagrams (1) and (2) in Figure 1 are graph pushouts. (See Definition A.1 in the Appendix for the definition of graph pushouts.) The notation $G \Rightarrow_{p,g} H$ is used when we wish to make $g : L \to G$ explicit.

The application of a rule $p = \langle L \leftarrow K \rightarrow R \rangle$ to a graph G, that is, the construction of the double-pushout in Figure 1, amounts to the following steps (see Ehrig (1979)):

(1) Find a graph morphism $g : L \to G$ and check the following two conditions.
Dangling condition: No edge in $G - g(L)$ is incident to a node in $g(L) - g(K)$.
Identification condition: For all distinct items $x, y \in L$, we have $g(x) = g(y)$ 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 $K \to D$ and $K \to R$, yielding a graph H and graph morphisms $D \to H$ and $R \to H$. (See Lemma A.2 in the Appendix for the construction of graph pushouts.)

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

3. Four variants of the double-pushout 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 $\mathrm{DPO}^{\mathrm{a/i}}$ the application of a rule p to a graph G is simpler than in $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$, in that step (1) can be replaced by step (1′):

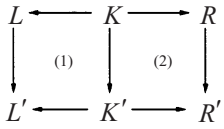
(1′) Find an injective graph morphism $g : L \to G$ and check the dangling condition.

This is more efficient than searching for an arbitrary matching morphism and checking the dangling as well as the identification condition. (When we simulate $\mathrm{DPO}^{\mathrm{x/a}}$ by $\mathrm{DPO}^{\mathrm{x/i}}$, however, this advantage will be offset by a greater number of rules.)

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.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

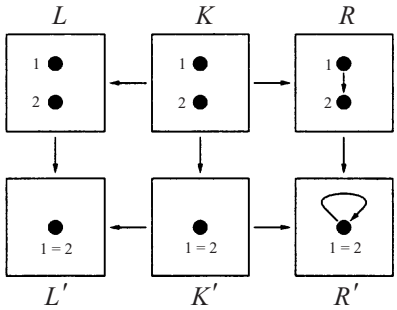
Definition 3.1. (Quotient rule) Given a rule $p = \langle L \leftarrow K \rightarrow R \rangle$, a rule $p' = \langle L' \leftarrow K' \rightarrow R' \rangle$ is a *quotient rule* of p if there are two graph pushouts of the form



where the vertical graph morphisms are surjective. The set of quotient rules of p is denoted by $\mathrm{Q}(p)$.

Since a rule has – up to isomorphism – only finitely many quotients, 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.

Example 3.2. Consider the rules $p = \langle L \leftarrow K \rightarrow R \rangle$ and $p' = \langle L' \leftarrow K' \rightarrow R' \rangle$ below. Then p' is a quotient rule of p. The set of quotient rules of p consists of the two non-isomorphic quotient rules p and p'.



In $\mathrm{DPO}^{\mathrm{i/i}}$, applying p means to choose two distinct nodes and to insert an edge between them. In $\mathrm{DPO}^{\mathrm{i/a}}$, these nodes need not be distinct and hence loops can be created.

Lemma 3.3. (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′) in Figure 2 on the left are graph pushouts. By the Composition Lemma (see Lemma A.18 in the Appendix), 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 $L \rightarrow L' \rightarrow G$ be an epi-mono factorization of $g : L \rightarrow G$, that is, $L \rightarrow L'$ is a surjective morphism and $L' \rightarrow G$ is an injective morphism such that $L \rightarrow G = L \rightarrow L' \rightarrow G$. This factorization induces a decomposition of the original diagrams into diagrams (1) and (1′), (2) and (2′) as follows. Construct K' as a pullback object of $L' \rightarrow G$ and $D \rightarrow G$, and denote the diagram by (1′). Then there is a unique graph morphism $K \rightarrow K'$ such that $K \rightarrow K' \rightarrow D = K \rightarrow D$ and such that diagram (1) in the right part of Figure 2 commutes. By the Decomposition Lemma A.19, (1) is a pullback. By the Pullback–Pushout Lemma A.12 and the Special Decomposition Lemma A.20, (1′) and (1) are pushouts. Moreover, the surjectivity of $L \rightarrow L'$ implies the surjectivity of

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

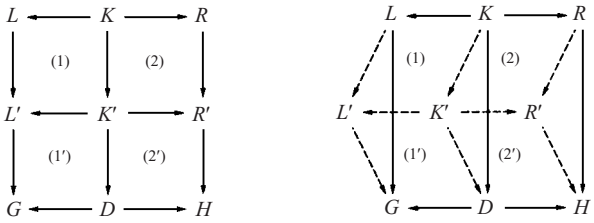


Fig. 2. Illustrations for the proof of the Quotient Lemma

$K \to K'$ (Surjectivity Lemma A.15). Now construct R' as the pushout object of $K \to K'$ and $K \to R$, and denote the diagram by (2). Then there is a unique graph morphism $R' \to H$ such that $R \to R' \to H = R \to H$ and such that diagram (2′) commutes. By the Decomposition Lemma A.19, diagram (2′) is a pushout. Moreover, the surjectivity of $K \to K'$ implies the surjectivity of $R \to R'$ (Surjectivity Lemma A.15). Therefore, all diagrams in the right part of Figure 2 are pushouts, and the vertical morphisms $L \to L'$, $K \to K'$ and $R \to R'$ are surjective. Without loss of generality, we may assume that $K' \to L'$ is an inclusion. Therefore, $p' = \langle L' \leftarrow K' \to R' \rangle$ is a quotient rule of p and $G \Rightarrow_{p',g'} H$ is a direct derivation through p' using an injective matching morphism g'. □

Theorem 3.4. (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. The theorem is an immediate consequence of Lemma 3.3. □

4. 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 non-terminal labels and show that these grammars can generate more languages in $\mathrm{DPO}^{x/\mathrm{i}}$ 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 $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$ more functions can be computed than in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$.

4.1. *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 non-terminal labels. In particular, we prove that there is a grammar without non-terminal labels in $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$ generating a language that cannot be generated by any grammar without non-terminal labels in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$.

Definition 4.1. (Graph grammar and graph language) Let $x, y \in \{\mathrm{a}, \mathrm{i}\}$. A *graph grammar* in $\mathrm{DPO}^{x/y}$ is a system $\mathscr{G} = \langle \mathscr{C}, \mathscr{R}, S, \mathscr{T} \rangle$, where $\mathscr{C}$ is a finite label alphabet, $\mathscr{T}$ is an alphabet of *terminal* labels with $\mathscr{T}_V \subseteq \mathscr{C}_V$ and $\mathscr{T}_E \subseteq \mathscr{C}_E$, S is a graph over $\mathscr{C}$ called

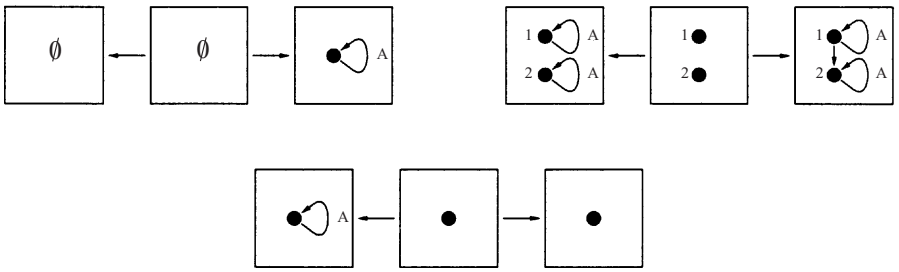
https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

the *start graph*, and $\mathscr{R}$ is a finite set of rules over $\mathscr{C}$ such that if $x = \mathrm{i}$, all rules in $\mathscr{R}$ are injective. The *graph language generated by* $\mathscr{G}$ is the set $\mathrm{L}(\mathscr{G})$ consisting of all graphs G over $\mathscr{T}$ such that there is a derivation $S \Rightarrow^*_{\mathscr{R}} G$ in $\mathrm{DPO}^{x/y}$. We use $\mathscr{L}^{x/y}$ to denote the class of all graph languages generated by graph grammars in $\mathrm{DPO}^{x/y}$.

Example 4.2. The set of all (directed, unlabelled) loop-free graphs can be generated from the empty graph by the following two rules if injective matching is assumed:



In the traditional double-pushout approach, this graph class can be generated from the empty graph by the following three rules using 'A' as a non-terminal label:



Note that the rule generating an unlabelled edge cannot be applied to a single node with a single loop. This is because the identification condition for direct derivations prevents the two loops in the left-hand side of the rule being identified. The idea used in this grammar allows us to show that arbitrary grammars in $\mathrm{DPO}^{x/\mathrm{i}}$ can be simulated in $\mathrm{DPO}^{x/\mathrm{a}}$. This will be done in the proof of the second part of the following lemma.

Lemma 4.3. Let $x \in \{\mathrm{a}, \mathrm{i}\}$.

(1) For every graph grammar $\mathscr{G}$ in $\mathrm{DPO}^{x/\mathrm{a}}$ there exists a graph grammar $\mathscr{G}'$ in $\mathrm{DPO}^{x/\mathrm{i}}$ such that $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$.

(2) For every graph grammar $\mathscr{G}$ in $\mathrm{DPO}^{x/\mathrm{i}}$ there exists a graph grammar $\mathscr{G}'$ in $\mathrm{DPO}^{x/\mathrm{a}}$ such that $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$.

Proof.

(1) Let $\mathscr{G} = \langle \mathscr{C}, \mathscr{R}, S, \mathscr{T} \rangle$ be a graph grammar in $\mathrm{DPO}^{x/\mathrm{a}}$ and $\mathrm{Q}(\mathscr{R}) = \bigcup_{p \in \mathscr{R}} \mathrm{Q}(p)$. Then $\mathrm{Q}(\mathscr{G}) = \langle \mathscr{C}, \mathrm{Q}(\mathscr{R}), S, \mathscr{T} \rangle$ is a graph grammar in $\mathrm{DPO}^{x/\mathrm{i}}$ such that, by the Simulation Theorem, $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathrm{Q}(\mathscr{G}))$.

(2) Let $\mathscr{G} = \langle \mathscr{C}, \mathscr{R}, S, \mathscr{T} \rangle$ be a graph grammar in $\mathrm{DPO}^{x/\mathrm{i}}$. Then we construct a graph grammar $\mathscr{G}' = \langle \mathscr{C}', \mathscr{R}', S', \mathscr{T} \rangle$ in $\mathrm{DPO}^{x/\mathrm{a}}$ according to the following idea. The rules of the original graph grammar are modified in such a way that every matching morphism that identifies nodes or edges violates the identification condition. Every node in the start graph, the left- and the right-hand sides is equipped with a loop; as a consequence, every matching morphism identifying two nodes must identify the incident loops that are

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

not present in the interface. Moreover, all edges are removed from the interfaces; as a consequence, every matching morphism identifying two edges violates the identification condition. More precisely, given a graph G, let $G^{\oplus}$ be the graph in which all nodes are equipped with an A-labelled loop such that $A \notin \mathscr{C}_E$ is a new edge label. Given a graph H, let $H^{\ominus}$ be the graph in which all edges are removed. For a rule $p = \langle L \leftarrow K \rightarrow R \rangle$, let $p^{\oplus}$ be the rule $\langle L^{\oplus} \leftarrow K^{\ominus} \rightarrow R^{\oplus} \rangle$. Finally, for a node label $a \in \mathscr{C}_V$, let $\mathrm{s}(a)$ be the graph with a single a-labelled node and $\mathrm{del}(a)$ be the rule $\langle \mathrm{s}(a)^{\oplus} \leftarrow \mathrm{s}(a) \rightarrow \mathrm{s}(a) \rangle$, which allows us to delete the equipped loop.

Now let $\mathscr{C}' = \langle \mathscr{C}_V, \mathscr{C}_E + \{A\} \rangle$, $\mathscr{R}' = \{p^{\oplus} \mid p \in \mathscr{R}\} \cup \{\mathrm{del}(a) \mid a \in \mathscr{C}_V\}$ and $S' = S^{\oplus}$. Then it is easy to see that for all graphs G and H, and every rule p, we have $G \Rightarrow_p H$ in $\mathrm{DPO}^{x/\mathrm{i}}$ if and only if $G^{\oplus} \Rightarrow_{p^{\oplus}} H^{\oplus}$ in $\mathrm{DPO}^{x/\mathrm{a}}$. Moreover, for all graphs G over $\mathscr{T}$, there is a derivation $S \Rightarrow^*_{\mathscr{R}} G$ in $\mathrm{DPO}^{x/\mathrm{i}}$ if and only if there is a derivation $S^{\oplus} \Rightarrow^*_{\mathscr{R}'} G$ in $\mathrm{DPO}^{x/\mathrm{a}}$. Hence $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$. □

By Lemma 4.3, we know that injective matching does not change the generative power of unrestricted graph grammars. The next lemma shows that going from injective rules to arbitrary, possibly non-injective, rules does not change the generative power either (in the case of injective matching).

Lemma 4.4. For every graph grammar $\mathscr{G}$ in $\mathrm{DPO}^{\mathrm{a/i}}$ there exists a graph grammar $\mathscr{G}'$ in $\mathrm{DPO}^{\mathrm{i/i}}$ such that $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$.

Proof. Let $\mathscr{G}$ be a graph grammar in $\mathrm{DPO}^{\mathrm{a/i}}$. We will construct a grammar $\mathscr{G}'$ in $\mathrm{DPO}^{\mathrm{i/i}}$ that simulates the identification of two edges by deleting one of the edges, and the identification of two nodes by redirecting all edges incident with one node to the other and then deleting the first one.

Simulation of edge identification. Let $\mathscr{G}$ be a graph grammar in $\mathrm{DPO}^{\mathrm{a/i}}$ with rule set $\mathscr{R}$. We modify $\mathscr{G}$ to a graph grammar $\mathscr{G}'$ by replacing $\mathscr{R}$ with $\mathscr{R}' = \{p' \mid p \in \mathscr{R}\}$, where $p' = \langle L \leftarrow K' \rightarrow^{r'} R \rangle$ is obtained from $p = \langle L \leftarrow K \rightarrow^{r} R \rangle$ as follows. Let K' be the subgraph of K with $V_{K'} = V_K$ and $E_{K'} = \{\mathrm{rep}([e]) \mid e \in E_K\}$, where $[e]$ is the equivalence class $\{e' \in E_K \mid r_E(e) = r_E(e')\}$ and $\mathrm{rep}([e])$ is a fixed element in $[e]$. Let $K' \rightarrow L = K' \rightarrow K \rightarrow L$ and $K' \rightarrow^{r'} R = K' \rightarrow K \rightarrow^{r} R$, where $K' \rightarrow K$ is the inclusion of K' in K. Then r'_E is injective. For all graphs G and H, we have $G \Rightarrow_{p,g} H$ in $\mathrm{DPO}^{\mathrm{a/i}}$ if and only if $G \Rightarrow_{p',g} H$ in $\mathrm{DPO}^{\mathrm{a/i}}$. Hence $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$.

Simulation of node identification. Let $\mathscr{G} = \langle \mathscr{C}, \mathscr{R}, S, \mathscr{T} \rangle$ be a graph grammar in $\mathrm{DPO}^{\mathrm{a/i}}$. According to the above construction, we may assume that for every rule $p = \langle L \leftarrow K \rightarrow^{r} R \rangle$ in $\mathscr{R}$, r_E is injective. We construct a graph grammar $\mathscr{G}'$ in $\mathrm{DPO}^{\mathrm{i/i}}$ as follows.

For every rule $p = \langle L \leftarrow K \rightarrow^{r} R \rangle$ we construct an injective rule $p' = \langle L \leftarrow K \rightarrow^{r'} R' \rangle$ by 'splitting' nodes and inserting edges with label 'eq' indicating that the adjacent nodes have to be equated. More precisely, let R' be the graph with node set $(V_R - r_V(V_K)) + V_K$ and edge set $E_R + \{\langle v, \mathrm{rep}([v]) \rangle \mid v \in V_K,\ v \neq \mathrm{rep}([v])\}$ where $[v]$ for $v \in V_K$ is the equivalence class $\{v' \in V_K \mid r_V(v) = r_V(v')\}$ and $\mathrm{rep}([v])$ is a fixed element in $[v]$. For a node $v \in V_{R'}$, we have $l_{R'}(v) = l_R(v)$ if $v \in V_R - r_V(V_K)$ and $l_{R'}(v) = l_K(v)$ otherwise. For an edge $e \in E_R$, we have $s_{R'}(e) = \mathrm{rep}([v])$ if $s_R(e) = r_V(v)$ for some $v \in V_K$ and

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

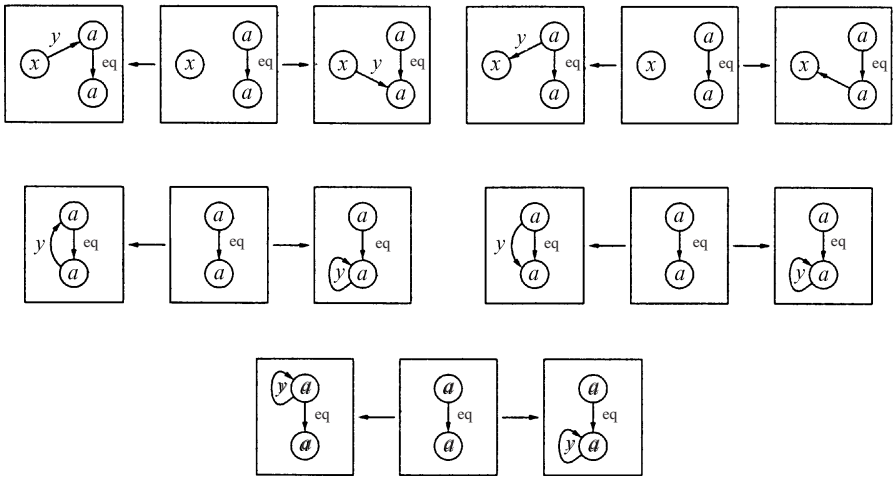


Fig. 3. Rules for redirecting edges

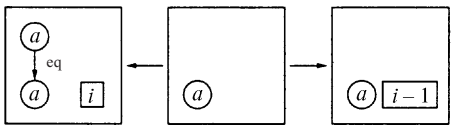


Fig. 4. Rules for deleting eq-labelled edges

$s_{R'}(e) = s_R(e)$ otherwise, $t_{R'}(e)$ is defined analogously, and $m_{R'}(e) = m_R(e)$. Moreover, an edge of the form $\langle v, \mathrm{rep}([v])\rangle$ has source v, target $\mathrm{rep}([v])$ and label eq, where $\mathrm{eq} \notin \mathscr{C}_E$ is a new non-terminal symbol. Let $r' : K \to R'$ be the morphism with $r'_V(v) = v$ and $r'_E(e) = r_E(e)$ for $v \in V_K$ and $e \in E_K$. Note that r' is injective. Let $\mathscr{R}' = \{p' \mid p \in \mathscr{R}\}$.

Next, we restrict the applicability of the rules to graphs without eq-labelled edges as follows. The start graph S is equipped with a node with label 0 indicating the absence of eq-labelled edges. Each rule p' in $\mathscr{R}'$ is equipped with additional nodes, where the application condition 'there do not exist eq-labelled edges' is represented by a node with label 0 on the left-hand side and the number of created eq-labelled edges is represented by a node with this number on the right-hand side. More formally, given a graph G and a non-negative integer i, $G[i]$ is the graph obtained from G by adding a single node with label i. For a rule $p' = \langle L \leftarrow K \to R'\rangle$ in $\mathscr{R}'$, let p'' be the rule $\langle L[0] \leftarrow K \to R'[i]\rangle$ where i is the number of eq-labelled edges in R'. The set $\{p'' \mid p' \in \mathscr{R}'\}$ is denoted by $\mathscr{R}''$. Moreover, let del be the rule for deleting a single node with label 0.

Finally, we add to $\mathscr{R}''$ rules for redirecting edges as shown in Figure 3, and rules for deleting eq-labelled edges as shown in Figure 4. The redirecting rules do not change the number of eq-labelled edges. The eq-deleting rules in Figure 4 are of the form $\langle L[i] \leftarrow K \to R[i-1]\rangle$ where $i \in \{1, \ldots, k\}$ and k is the maximal number of eq-labelled edges in a right-hand side of $\mathscr{R}''$. The set of all redirecting edges with $a, x \in \mathscr{C}_V$ and $y \in \mathscr{C}_E$, and all deleting rules is denoted by EQ.

Now let $\mathscr{G}' = \langle \mathscr{C}', \mathscr{R}''', S', \mathscr{T}\rangle$ be the graph grammar in $\mathrm{DPO}^{\mathrm{i/i}}$ with $\mathscr{C}' = \langle \mathscr{C}_V +$

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

$\{0,\ldots,k\}, \mathscr{C}_E + \{\mathrm{eq}\}\rangle$, $\mathscr{R}''' = \mathscr{R}'' \cup \mathrm{EQ} \cup \{\mathrm{del}\}$ and $S' = S[0]$. Then it is easy to see that for all graphs G and H over $\mathscr{C}$ and each rule $p \in \mathscr{R}$, we have $G \Rightarrow_p H$ in $\mathrm{DPO}^{\mathrm{a/i}}$ if and only if $G[0] \Rightarrow_{p''} H'[i] \Rightarrow^*_{\mathrm{EQ}} H[0]$ in $\mathrm{DPO}^{\mathrm{i/i}}$ for some $i \in \{0,\ldots,k\}$. Moreover, for all graphs G over $\mathscr{T}$, we have $S \Rightarrow^*_{\mathscr{R}} G$ in $\mathrm{DPO}^{\mathrm{a/i}}$ if and only if $S[0] \Rightarrow^*_{\mathscr{R}'''} G$ in $\mathrm{DPO}^{\mathrm{i/i}}$. This may be seen as follows. For every derivation $S \Rightarrow^*_{\mathscr{R}} G$ there is a derivation $S[0] \Rightarrow^*_{\mathscr{R}'' \cup \mathrm{EQ}} G[0] \Rightarrow_{\mathrm{del}} G$. For the opposite direction, every derivation $S[0] \Rightarrow^*_{\mathscr{R}'''} G$ to a graph G over $\mathscr{T}$ is of the form $S[0] = G_0[0] \Rightarrow_{p_1''} G_1'[i_1] \Rightarrow^*_{\mathrm{EQ}} G_1[0] \Rightarrow_{p_2''} \ldots \Rightarrow_{p_n''} G_n'[i_n] \Rightarrow^*_{\mathrm{EQ}} G_n[0] \Rightarrow_{\mathrm{del}} G$. As a consequence, there is a derivation $S = G_0 \Rightarrow_{p_1} G_1 \Rightarrow_{p_2} \ldots \Rightarrow_{p_n} G_n = G$. Hence $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$. □

An alternative proof of Lemma 4.4 can be obtained from Uesu's result that $\mathscr{L}^{\mathrm{i/i}}$ is the class of all recursively enumerable sets of graphs (Uesu 1978). (A set of graphs over some label alphabet is recursively enumerable if its image under a Gödel numbering is a recursively enumerable set of natural numbers.) This is because each graph language in $\mathscr{L}^{\mathrm{a/i}}$ is clearly recursively enumerable and hence $\mathscr{L}^{\mathrm{a/i}} \subseteq \mathscr{L}^{\mathrm{i/i}}$. However, our proof also shows how to transform an arbitrary grammar in $\mathrm{DPO}^{\mathrm{a/i}}$ into an equivalent grammar in $\mathrm{DPO}^{\mathrm{i/i}}$.

Theorem 4.5. (Generative power I) $\mathscr{L}^{\mathrm{i/a}} = \mathscr{L}^{\mathrm{i/i}} = \mathscr{L}^{\mathrm{a/a}} = \mathscr{L}^{\mathrm{a/i}}$.

Proof. By Lemma 4.3, we have $\mathscr{L}^{x/\mathrm{a}} \subseteq \mathscr{L}^{x/\mathrm{i}} \subseteq \mathscr{L}^{x/\mathrm{a}}$ for $x \in \{\mathrm{a},\mathrm{i}\}$. Hence $\mathscr{L}^{\mathrm{i/a}} = \mathscr{L}^{\mathrm{i/i}}$ and $\mathscr{L}^{\mathrm{a/a}} = \mathscr{L}^{\mathrm{a/i}}$. By Lemma 4.4 and the fact that every graph grammar in $\mathrm{DPO}^{\mathrm{i/i}}$ is also in $\mathrm{DPO}^{\mathrm{a/i}}$, we obtain $\mathscr{L}^{\mathrm{a/i}} \subseteq \mathscr{L}^{\mathrm{i/i}} \subseteq \mathscr{L}^{\mathrm{a/i}}$. Thus $\mathscr{L}^{\mathrm{i/a}} = \mathscr{L}^{\mathrm{i/i}} = \mathscr{L}^{\mathrm{a/i}} = \mathscr{L}^{\mathrm{a/a}}$. □

Next we put Theorem 4.5 into perspective by showing that injective matching provides more generative power if we restrict ourselves to grammars without non-terminal labels. To this end, we use $\mathscr{L}_{\mathrm{T}}^{x/y}$ to denote the class of all graph languages generated by a grammar $\langle \mathscr{C}, \mathscr{R}, S, \mathscr{T}\rangle$ in $\mathrm{DPO}^{x/y}$ with $\mathscr{T} = \mathscr{C}$. Note that for such a grammar, every graph derivable from S belongs to the generated language.

Theorem 4.6. (Generative power II) For $x \in \{\mathrm{a},\mathrm{i}\}$, $\mathscr{L}_{\mathrm{T}}^{x/\mathrm{a}} \subset \mathscr{L}_{\mathrm{T}}^{x/\mathrm{i}}$.

Proof. By the proof of Lemma 4.3, for every grammar $\mathscr{G}$ in $\mathrm{DPO}^{x/\mathrm{a}}$ without non-terminal labels there is a grammar $\mathscr{G}'$ in $\mathrm{DPO}^{x/\mathrm{i}}$ without non-terminal labels such that $\mathrm{L}(\mathscr{G}) = \mathrm{L}(\mathscr{G}')$. Hence $\mathscr{L}_{\mathrm{T}}^{x/\mathrm{a}} \subseteq \mathscr{L}_{\mathrm{T}}^{x/\mathrm{i}}$.

To show that the inclusion is strict, let $\mathscr{G} = \langle \mathscr{C}, \mathscr{R}, S, \mathscr{T}\rangle$ be the grammar in $\mathrm{DPO}^{\mathrm{i/i}}$ defined by S is the empty graph, $\mathscr{C}_{\mathrm{V}}$ and $\mathscr{C}_{\mathrm{E}}$ are singletons, $\mathscr{T} = \mathscr{C}$, and $\mathscr{R}$ consists of the following two rules (which have already appeared in the introduction and in Example 4.2):



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

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

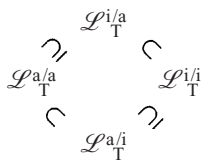
First we show that rules that identify nodes cannot occur in derivations of graphs in $\mathrm{L}(\mathscr{G}')$. Let $p = \langle L \leftarrow K \rightarrow^{r} R \rangle$ be a rule in $\mathscr{R}'$ such that $r_{\mathrm{V}}(v) = r_{\mathrm{V}}(v')$ for two distinct nodes v and v' in K. Then L cannot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

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

Let k be the maximal number of nodes occurring in S' or in a right-hand side of $\mathscr{R}'$. Consider a loop-free graph G with $k+1$ nodes such that there is an edge between each two distinct nodes. Let $S' = G_0 \Rightarrow_{\mathscr{R}'} G_1 \Rightarrow_{\mathscr{R}'} \ldots \Rightarrow_{\mathscr{R}'} G_n \cong G$ be a derivation generating G, and $i \in \{0, \ldots, n-1\}$ be the largest index such that $G_i \Rightarrow_{\mathscr{R}'} G_{i+1}$ creates a node v that is not removed in the rest of the derivation $G_{i+1} \Rightarrow^{*}_{\mathscr{R}'} G_n$. Then $\mathrm{V}_G = \mathrm{V}_{G_n}$ is contained in $\mathrm{V}_{G_{i+1}}$ up to isomorphism. Without loss of generality, 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 $G_i \Rightarrow_{\mathscr{R}'} G_{i+1}$, 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 $G_i \Rightarrow_{\mathscr{R}'} G_{i+1}$, there is no edge between v and v' in G_{i+1}. As there is an edge between v and v' in G, and $G_{i+1} \Rightarrow^{*}_{\mathscr{R}'} G$ does not identify nodes, there is a step $G_j \Rightarrow_{\mathscr{R}'} G_{j+1}$ with $j \geqslant i+1$ 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 $G_j \Rightarrow_{\mathscr{R}'} G_{j+1}$. 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 $g : L \rightarrow \tilde{L}$ 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 $\mathrm{L}(\mathscr{G}')$. □

By Theorem 4.6 and the fact that $\mathrm{DPO}^{\mathrm{i}/y}$ is contained in $\mathrm{DPO}^{\mathrm{a}/y}$, for $y \in \{\mathrm{a}, \mathrm{i}\}$, we have the following relations between the four classes of graph languages generated by grammars without non-terminal labels:



It is open whether or not the inclusions $\mathscr{L}^{\mathrm{i}/\mathrm{a}}_{\mathrm{T}} \subseteq \mathscr{L}^{\mathrm{a}/\mathrm{a}}_{\mathrm{T}}$ and $\mathscr{L}^{\mathrm{i}/\mathrm{i}}_{\mathrm{T}} \subseteq \mathscr{L}^{\mathrm{a}/\mathrm{i}}_{\mathrm{T}}$ are also strict.

4.2. *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. In this subsection we show that injective matching provides additional power for computing

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

functions on graphs by convergent graph transformation systems: we will prove that in $\mathrm{DPO}^{x/\mathrm{i}}$ more functions can be computed than in $\mathrm{DPO}^{x/\mathrm{a}}$, for $x \in \{\mathrm{a}, \mathrm{i}\}$.

Since the result of applying a rule in the double-pushout approach is unique only up to isomorphism, we consider derivations and functions on isomorphism classes of graphs. An *abstract graph* over a label alphabet $\mathscr{C}$ is an isomorphism class of graphs over $\mathscr{C}$. We write $[G]$ for the isomorphism class of a graph G, and $\mathscr{A}_{\mathscr{C}}$ for the set of all abstract graphs over $\mathscr{C}$.

Definition 4.7. (Graph transformation system) A *graph transformation system* in $\mathrm{DPO}^{x/y}$, for $x, y \in \{\mathrm{a}, \mathrm{i}\}$, is a pair $\langle \mathscr{C}, \mathscr{R} \rangle$ where $\mathscr{C}$ is a label alphabet and $\mathscr{R}$ a set of rules with graphs over $\mathscr{C}$ such that if $x = \mathrm{i}$, then all rules are injective.

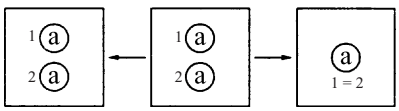
In other words, a graph transformation system is obtained from a graph grammar by forgetting the start graph and the non-terminal labels. Such a system $\langle \mathscr{C}, \mathscr{R} \rangle$ is *finite* if $\mathscr{C}_{\mathrm{V}}$, $\mathscr{C}_{\mathrm{E}}$ and $\mathscr{R}$ are finite sets. We will often identify $\langle \mathscr{C}, \mathscr{R} \rangle$ with $\mathscr{R}$, leaving $\mathscr{C}$ implicit. The derivation relation $\Rightarrow_{\mathscr{R}}$ on graphs over $\mathscr{C}$ (defined in Section 2) is lifted to $\mathscr{A}_{\mathscr{C}}$ by

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

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

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

Example 4.8. Let $\mathscr{R}$ consist of the following single rule (also shown in the introduction):



This system is convergent in $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$: termination follows from the fact that every direct derivation reduces the number of nodes in a graph; confluence is a consequence of the fact that for every constellation $[G_1] \Leftarrow_{\mathscr{R}} [G] \Rightarrow_{\mathscr{R}} [G_2]$, there is a graph H such that $[G_1] \Rightarrow_{\mathscr{R}} [H] \Leftarrow_{\mathscr{R}} [G_2]$, or $[G_1] = [G_2]$. (See Huet (1980) or Baader and Nipkow (1998) for a proof that a binary relation with the latter property is confluent.) The function $\mathrm{N}_{\mathscr{R}}$ sends every abstract graph $[G]$ to $[G']$, where G' is obtained from G by merging all a-labelled nodes into a single node.

One should note that $\mathscr{R}$ is not convergent in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$ because it admits the following

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

infinite derivation sequence:

$$\textcircled{a} \Rightarrow_{\mathscr{R}} \textcircled{a} \Rightarrow_{\mathscr{R}} \ldots$$

A discussion of general criteria for proving termination and confluence of graph transformation systems is beyond the scope of this paper. For $\mathrm{DPO}^{a/a}$, a condition for termination in terms of so-called forward closures is given in Plump (1995) and a condition for confluence in terms of so-called critical pairs can be found in Plump (1993).

Definition 4.9. (DPO-computable function) Let $x, y \in \{a, i\}$. A function $f : \mathscr{A}_{\mathscr{C}} \to \mathscr{A}_{\mathscr{C}}$ is *computable* in $\mathrm{DPO}^{x/y}$, or $\mathrm{DPO}^{x/y}$*-computable* for short, if there exists a finite, convergent graph transformation system $\langle \mathscr{C}, \mathscr{R} \rangle$ in $\mathrm{DPO}^{x/y}$ such that $\mathrm{N}_{\mathscr{R}} = f$. The class of all $\mathrm{DPO}^{x/y}$-computable functions is denoted by $\mathscr{F}^{x/y}$.

Note that not all computable functions on graphs are DPO-computable. See Habel and Plump (2001) for a universal computability model based on programs over double-pushout rules.

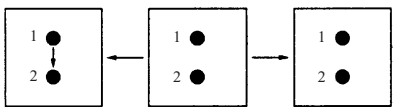
Theorem 4.10. (Computability I) $\mathscr{F}^{x/a} \subseteq \mathscr{F}^{x/i}$, for $x \in \{a, i\}$.

Proof. Let $\langle \mathscr{C}, \mathscr{R} \rangle$ be a finite, convergent graph transformation system in $\mathrm{DPO}^{x/a}$. Then $\langle \mathscr{C}, \mathrm{Q}(\mathscr{R}) \rangle$ is a finite and convergent system in $\mathrm{DPO}^{x/i}$, 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 $\mathscr{R}$ and $\mathrm{Q}(\mathscr{R})$, respectively. Thus $\mathrm{N}_{\mathscr{R}} = \mathrm{N}_{\mathrm{Q}(\mathscr{R})}$, 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. Our next result shows that there is a function that is computable in $\mathrm{DPO}^{i/i}$ but not in $\mathrm{DPO}^{a/a}$. Since $\mathrm{DPO}^{i/i}$ is included in $\mathrm{DPO}^{a/i}$, and $\mathrm{DPO}^{i/a}$ is included in $\mathrm{DPO}^{a/a}$, the result at the same time shows that $\mathrm{DPO}^{i/i}$ is more powerful than $\mathrm{DPO}^{i/a}$ and that $\mathrm{DPO}^{a/i}$ is more powerful than $\mathrm{DPO}^{a/a}$.

Theorem 4.11. (Computability II) There exists a function that is computable in $\mathrm{DPO}^{i/i}$ but not in $\mathrm{DPO}^{a/a}$.

Proof. Let $\mathscr{C}$ be a label alphabet such that $\mathscr{C}_{\mathrm{V}}$ and $\mathscr{C}_{\mathrm{E}}$ are singletons. (This simplifies the reasoning but is not necessary for proving the theorem.) Let $f : \mathscr{A}_{\mathscr{C}} \to \mathscr{A}_{\mathscr{C}}$ be the function sending an abstract graph $[G]$ to $[G']$, where G' is obtained from G by removing all edges that are not loops. Obviously, f is computed by the convergent system $\mathscr{R}$ in $\mathrm{DPO}^{i/i}$ consisting of the following single rule:



To show that f cannot be computed in $\mathrm{DPO}^{a/a}$, suppose that $\mathscr{R}'$ is a finite, convergent graph transformation system in $\mathrm{DPO}^{a/a}$ such that $\mathrm{N}_{\mathscr{R}'} = f$. Let k be the maximal number of edges occurring in a left-hand side of a rule in $\mathscr{R}'$. Consider a graph G consisting of

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

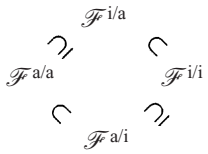
two nodes and $k+1$ edges between these nodes, and a derivation $[G] \Rightarrow^*_{\mathscr{R}'} f([G])$. Let $[G] \Rightarrow_{\mathscr{R}'} [H]$ be the first step of the derivation, with an underlying step $G \Rightarrow_{p,g} H$ on graphs. By the dangling condition, p cannot remove one of the two nodes in G. Next consider a graph G' consisting of a single node and $k+1$ loops, and a surjective graph morphism $h: G \to G'$. 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 $[G'] = f([G']) = \mathrm{N}_{\mathscr{R}'}([G'])$, and hence G' has to be a normal form with respect to $\mathscr{R}'$, which is a contradiction. □

Corollary 4.12. $\mathscr{F}^{\mathrm{i/a}} \subset \mathscr{F}^{\mathrm{i/i}}$ and $\mathscr{F}^{\mathrm{a/a}} \subset \mathscr{F}^{\mathrm{a/i}}$.

Proof. Combine Theorem 4.10 with Theorem 4.11. □

Example 4.13. Consider again the convergent one-rule system $\mathscr{R}$ of Example 4.8. One can show that $\mathrm{N}_{\mathscr{R}}$ is not computable in $\mathrm{DPO}^{\mathrm{a/a}}$ by the same argument that we used to prove Theorem 4.11. For, suppose there were a convergent system $\mathscr{R}'$ in $\mathrm{DPO}^{\mathrm{a/a}}$ such that $\mathrm{N}_{\mathscr{R}'} = \mathrm{N}_{\mathscr{R}}$. This system must contain a rule r that is applicable to a graph consisting of two a-labelled nodes and $k+1$ edges between these nodes, where k is the maximum number of edges occurring in the left-hand side of any rule in $\mathscr{R}'$. Then r does not remove nodes and hence is also applicable to the graph consisting of a single a-labelled node and $k+1$ loops, contradicting the fact that this graph must be a normal form with respect to $\mathscr{R}'$.

The following diagram summarizes Theorem 4.10 and Corollary 4.12:



We conjecture that the inclusions $\mathscr{F}^{\mathrm{i/a}} \subseteq \mathscr{F}^{\mathrm{a/a}}$ and $\mathscr{F}^{\mathrm{i/i}} \subseteq \mathscr{F}^{\mathrm{a/i}}$ are strict, too. However, this problem appears to be much harder than the one solved by Theorem 4.11.

5. Parallel independence

In this section we consider pairs of direct derivations $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ and look for conditions under which there are direct derivations $H_1 \Rightarrow_{p_2} M \Leftarrow_{p_1} H_2$ or $H_1 \Rightarrow_{\mathrm{Q}(p_2)} M \Leftarrow_{\mathrm{Q}(p_1)} H_2$. We formulate the notions of parallel and strong parallel independence and present three parallel commutativity results.

Notation. In this and the following four sections, let p_1 and p_2 be two rules with $p_i = \langle L_i \leftarrow K_i \rightarrow^{r_i} R_i \rangle$, for $i = 1, 2$.

Definition 5.1. (Parallel independence) Two direct derivations $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ as in Figure 5 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 $L_2 \to D_1 \to G = L_2 \to G$.

Roughly speaking, the direct derivations are parallelly independent if and only if the intersection of the left-hand sides of p_1 and p_2 in G consists of common interface items.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

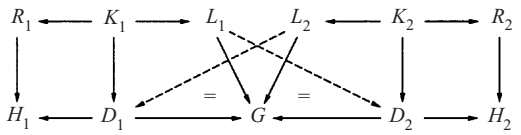
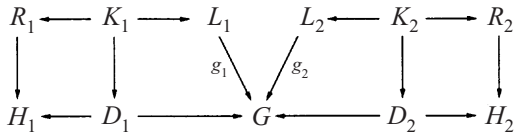


Fig. 5. Parallel independence

The notion of parallel independence and its characterization are well known, see Ehrig and Kreowski (1976), and Ehrig and Rosen (1976), respectively.

Lemma 5.2. (Characterization of parallel independence) For $x, y \in \{\mathrm{a}, \mathrm{i}\}$, two direct derivations $H_1 \Leftarrow_{p_1,g_1} G \Rightarrow_{p_2,g_2} H_2$ 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 interface items, that is, $g_1(L_1) \cap g_2(L_2) \subseteq g_1(K_1) \cap g_2(K_2)$.



Theorem 5.3. (Parallel commutativity I) In $\mathrm{DPO}^{\mathrm{i}/\mathrm{a}}$, $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$, and $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$, for every pair of parallelly independent direct derivations $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ there are two direct derivations of the form $H_1 \Rightarrow_{p_2} M \Leftarrow_{p_1} H_2$.

Proof. In Ehrig and Kreowski (1976) and Ehrig (1979), the statement is proved for direct derivations in $\mathrm{DPO}^{\mathrm{i}/\mathrm{a}}$ and $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$. For $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$, the statement follows by inspecting the proof: if the original derivations are in $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$, all morphisms occurring in the proof are injective. Consequently, the direct derivations $H_1 \Rightarrow_{p_2} M \Leftarrow_{p_1} H_2$ are in $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$. □

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

Example 5.4. The direct derivations in Figure 6 are parallelly independent, but no direct derivations of the form $H \Rightarrow_p M \Leftarrow_p H$ exist in $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$. The reason is that the composed morphism $L \to D \to H$ is not injective.

This counter-example suggests that we should strengthen parallel independence as follows.

Definition 5.5. (Strong parallel independence) Two direct derivations $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ as in Figure 5 are *strongly parallelly independent* if there are graph morphisms $L_1 \to D_2$ and $L_2 \to D_1$ such that the following hold:

1 $L_1 \to D_2 \to G = L_1 \to G$ and $L_2 \to D_1 \to G = L_2 \to G$.
2 $L_1 \to D_2 \to H_2$ and $L_2 \to D_1 \to H_1$ are injective.

Example 5.6. The direct derivations in Figure 7 are strongly parallelly independent and there exist direct derivations of the form $H_1 \Rightarrow_{p_2} M \Leftarrow_{p_1} H_2$ in $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

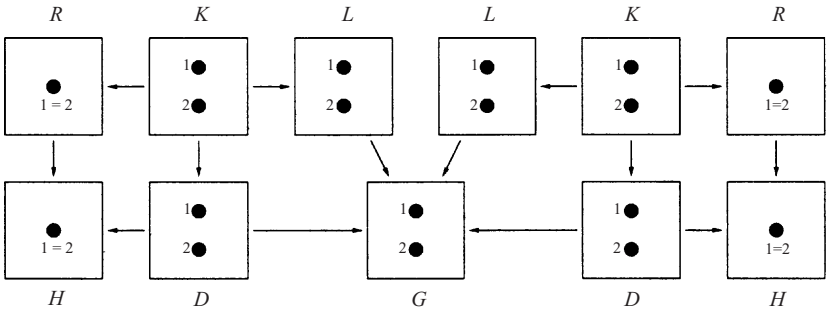


Fig. 6. A counter-example to parallel commutativity in DPO$^{a/i}$

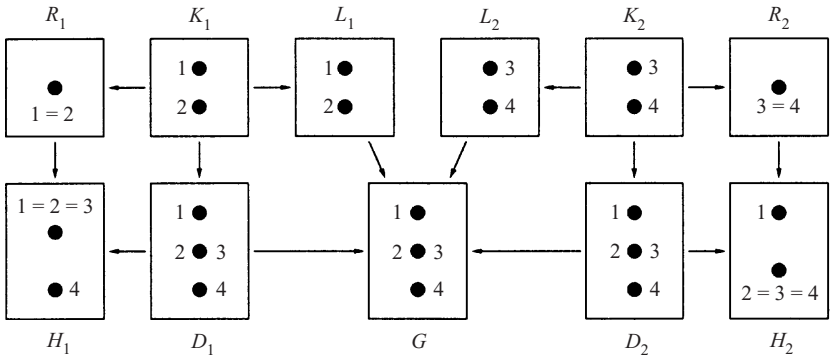


Fig. 7. Two strongly parallelly independent direct derivations

Theorem 5.7. (Parallel commutativity II) In DPO$^{a/i}$, for every pair of strongly parallelly independent direct derivations $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ there are two direct derivations of the form $H_1 \Rightarrow_{p_2} M \Leftarrow_{p_1} H_2$.

Proof. Let $H_1 \Leftarrow_{p_1,g_1} G \Rightarrow_{p_2,g_2} H_2$ in DPO$^{a/i}$ be strongly parallelly independent. Then the two steps are in DPO$^{a/a}$ and are parallelly independent. By the proof of the Parallel Commutativity Theorem I, there are direct derivations $H_1 \Rightarrow_{p_2,g_2'} M \Leftarrow_{p_1,g_1'} H_2$ in DPO$^{a/a}$ such that $g_2' = L_2 \to D_1 \to H_1$ and $g_1' = L_1 \to D_2 \to H_2$. Since both morphisms are injective by strong parallel independence, the derivations are in DPO$^{a/i}$. □

The next result is obtained by combining the Parallel Commutativity Theorem I with the Simulation Theorem.

Theorem 5.8. (Parallel commutativity III) In DPO$^{a/i}$, for every pair of parallelly independent direct derivations $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ there are two direct derivations of the form $H_1 \Rightarrow_{Q(p_2)} M \Leftarrow_{Q(p_1)} H_2$.

Proof. Let $H_1 \Leftarrow_{p_1} G \Rightarrow_{p_2} H_2$ in DPO$^{a/i}$ be parallelly independent. Then the derivations are in DPO$^{a/a}$, and, by the Parallel Commutativity Theorem I, there are direct derivations $H_1 \Rightarrow_{p_2} M \Leftarrow_{p_1} H_2$ in DPO$^{a/a}$. Hence, by the Simulation Theorem, there are direct derivations $H_1 \Rightarrow_{Q(p_2)} M \Leftarrow_{Q(p_1)} H_2$ in DPO$^{a/i}$. □

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

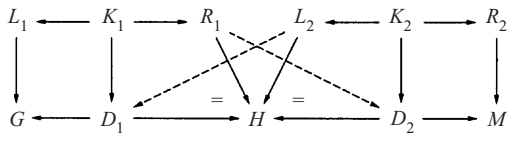


Fig. 8. Sequential independence

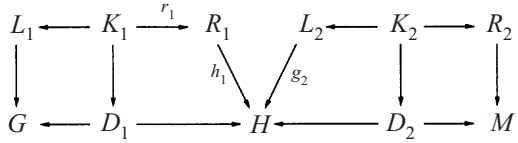
6. Sequential independence

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

Definition 6.1. (Sequential independence) Two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ as in Figure 8 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 $L_2 \to D_1 \to H = L_2 \to H$. In this case the pair $\langle R_1 \to D_2, L_2 \to D_1 \rangle$ is said to be an *independence pair* of $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$.

Analogously to the situation for parallel independence, if p_1 and p_2 are injective rules, the direct derivations are sequentially independent if and only if the intersection of the right-hand side of p_1 with the left-hand side of p_2 in H coincides with the intersection of the two interfaces. The characterization of sequential independence can be proved analogously to Lemma 5.2, see Ehrig and Rosen (1976).

Lemma 6.2. (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 interface items, that is, $h_1(R_1) \cap g_2(L_2) \subseteq h_1(r_1(K_1)) \cap g_2(K_2)$.



This characterization may break down, however, in the presence of non-injective rules. So Lemma 6.2 does not hold in $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$ and $\mathrm{DPO}^{\mathrm{a}/\mathrm{i}}$, as is demonstrated by the next example.

Example 6.3. Consider the direct derivations in Figure 9. We have $R_1 = r_1(K_1)$ and $L_2 = K_2$, so the condition of Lemma 6.2 is satisfied. But the two direct derivations are not sequentially independent because there is no graph morphism from L_2 to D_1.

Theorem 6.4. (Sequential commutativity I) In $\mathrm{DPO}^{\mathrm{i}/\mathrm{a}}$, $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$ and $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$, for every pair of sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ there are two sequentially independent direct derivations of the form $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$.

For $\mathrm{DPO}^{\mathrm{i}/\mathrm{a}}$ and $\mathrm{DPO}^{\mathrm{a}/\mathrm{a}}$, the statement is proved in Ehrig and Kreowski (1976). For $\mathrm{DPO}^{\mathrm{i}/\mathrm{i}}$, the statement follows by inspecting the proof. Since the proof has to be inspected a second time, it is given below.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

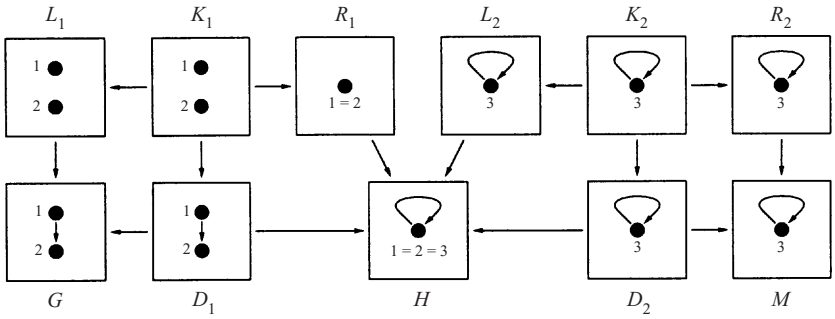
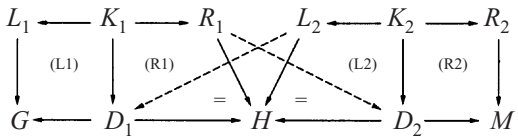


Fig. 9. A counter-example to sequential independence

Proof. Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be two sequentially independent direct derivations. Then we have the following situation:



To obtain a derivation $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$, we decompose the diagrams of the derivation $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$, rearrange them, and compose the diagrams in a new way.

Decomposition. Construct D_0 as pullback object of $D_1 \to H \leftarrow D_2$. Then there exist unique graph morphisms $K_i \to D_0$ $(i = 1, 2)$ such that $K_i \to D_0 \to D_i = K_i \to D_i$ $(i = 1, 2)$, $K_1 \to D_0 \to D_2 = K_1 \to R_1 \to D_2$ and $K_2 \to D_0 \to D_1 = K_2 \to L_2 \to D_1$ hold. Construct $\bar{D}_2$ and $\bar{D}_1$ as pushout objects of $L_1 \leftarrow K_1 \to D_0$ and $R_2 \leftarrow K_2 \to D_0$, respectively. Then there exist unique graph morphisms $\bar{D}_2 \to G$ and $\bar{D}_1 \to M$ such that $L_1 \to \bar{D}_2 \to G = L_1 \to G$ and $D_0 \to \bar{D}_2 \to G = D_0 \to D_1 \to G$, $R_2 \to \bar{D}_1 \to M = R_2 \to M$ and $D_0 \to \bar{D}_1 \to M = D_0 \to D_2 \to M$ hold.

Then all diagrams in Figure 10 are pushouts. This may be seen as follows. Since $K_2 \to L_2$ is assumed to be injective, $D_2 \to H$ and $D_0 \to D_1$ are injective (see Lemma A.14 in the Appendix). Since the diagrams (R1) and (L2) are pushouts, diagram (7) is constructed as pullback, and $D_0 \to D_1$ is injective, the diagrams (7), (R3) and (L4) are pushouts (see the Pullback–Pushout Lemma A.12 and the Special Decomposition Lemma A.20). Since the diagrams (L1) and (R2) are pushouts and the diagrams (L3) and (R4) are constructed as pushouts, the diagrams (L5) and (R6) are pushouts (see the Decomposition Lemma A.19).

Composition. Construct $\bar{H}$ as pushout object of $\bar{D}_1 \leftarrow D_0 \to \bar{D}_2$. Define $L_2 \to G$, $K_2 \to \bar{D}_2$, $R_2 \to \bar{H}$, $L_1 \to \bar{H}$, $K_1 \to \bar{D}_1$ and $R_1 \to M$ as the composed graph morphisms.

Then, the composed diagrams in Figure 11 are pushouts (Composition Lemma A.18). Therefore, there are direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$. From Figure 11 one can see that these steps are sequentially independent.

If the original derivations are in $\mathrm{DPO}^{a/a}$ ($\mathrm{DPO}^{i/a}$), the resulting derivations are in $\mathrm{DPO}^{a/a}$ ($\mathrm{DPO}^{i/a}$). If the original derivations are in $\mathrm{DPO}^{i/i}$, we have the following. Since the

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

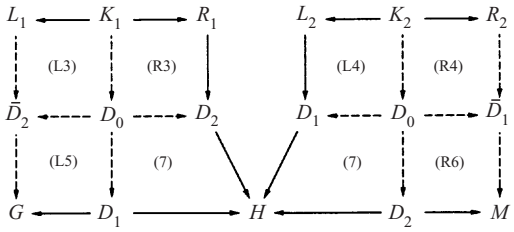


Fig. 10. Decomposition of a derivation

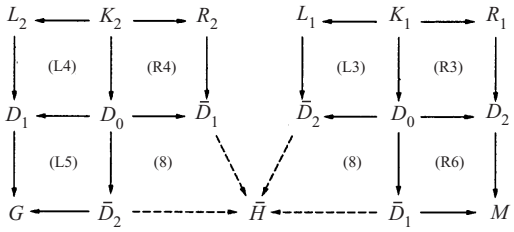


Fig. 11. Composition of a derivation

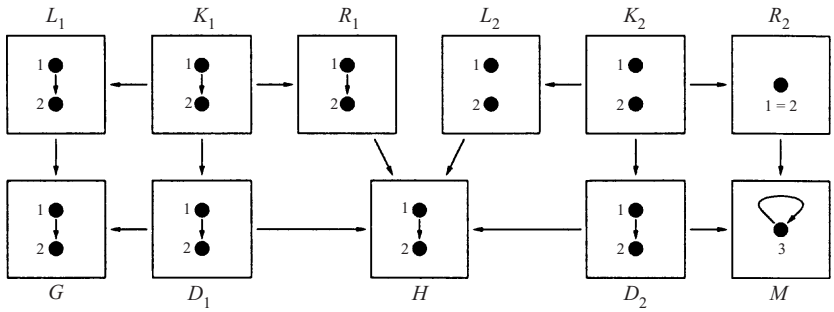


Fig. 12. A counter-example to sequential commutativity in DPO$^{a/i}$

rules are injective, all horizontal morphisms in the figures 10 and 11 are injective (Injectivity Lemma A.14); since the matching morphisms are injective, all vertical morphisms in the figures are injective (Pushout–Pullback-Lemma A.11, Injectivity Lemma A.14). Therefore, the two sequentially independent direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ are in DPO$^{i/i}$. □

The following counter-example demonstrates that in DPO$^{a/i}$, sequential independence does not guarantee sequential commutativity.

Example 6.5. The direct derivations in Figure 12 are sequentially independent. Moreover, there exists a direct derivation of the form $G \Rightarrow_{p_2} M$ in DPO$^{a/i}$. However, there is no step $M \Rightarrow_{p_1} M$ in DPO$^{a/i}$, since the composed morphism $R_1 \to D_2 \to M$ is not injective.

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, as this follows from the injectivity of $L_2 \to H$ and $K_1 \to L_1$.

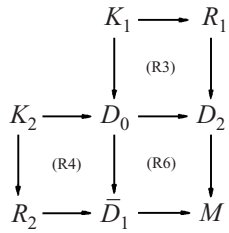
https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Definition 6.6. (Strong sequential independence) Two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ as in Figure 8 are *strongly sequentially independent* if there are graph morphisms $R_1 \to D_2$ and $L_2 \to D_1$ such that the following hold:

1 $R_1 \to D_2 \to H = R_1 \to H$ and $L_2 \to D_1 \to H = L_2 \to H$.
2 $R_1 \to D_2 \to M$ is injective.

Theorem 6.7. (Sequential commutativity II) In $\mathrm{DPO}^{a/i}$, for every pair of strongly sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ there are two strongly sequentially independent direct derivations of the form $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$.

Proof. Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{a/i}$ be strongly sequentially independent. Then the direct derivations are in $\mathrm{DPO}^{a/a}$ and are sequentially independent. By Theorem 6.4, there are two sequentially independent direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ in $\mathrm{DPO}^{a/a}$. Since the original derivation is in $\mathrm{DPO}^{a/i}$, the morphism $R_2 \to M = R_2 \to \bar{D}_1 \to M$ is injective, that is, $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ is strongly sequentially independent. It remains to show that the direct derivations are in $\mathrm{DPO}^{a/i}$. Inspecting the proof of Theorem 6.4 and looking at Figures 10 and 11, one may see the following. Since $L_2 \to H$ and $K_1 \to L_1$ are injective, $L_2 \to D_1$, $D_1 \to G$, and $L_2 \to G = L_2 \to D_1 \to G$ are injective. So it remains to show that $L_1 \to \bar{D}_2 \to \bar{H}$ is injective. Consider the following figure:



We will show that (R3)+(R6) is a pullback. By construction, (R3)+(R6) is a pushout. By assumption, $R_1 \to D_2 \to M$ is injective. Hence, according to the Pushout–Pullback Lemma A.11, it suffices to show that the pair $\langle K_1 \to R_1, K_1 \to D_0 \to \bar{D}_1 \rangle$ is monomorphic. Suppose to the contrary, that there are two distinct items x and y in K_1 that are identified by both $K_1 \to R_1$ and $K_1 \to D_0 \to \bar{D}_1$. Let x' and y' be the images of x and y in D_0, and z the image of x' and y' in D_2. ($D_0 \to D_2$ identifies x' and y' by the commutativity of (R3).) Since $K_1 \to D_0$ is injective, x' and y' are distinct. Since $D_0 \to \bar{D}_1$ identifies x' and y', and (R4) is a pushout, x' and y' are the images of two items in K_2 (Pushout Characterization A.7). But then $K_2 \to D_0 \to D_2$ sends these items to z, contradicting the injectivity of this morphism. Thus (R3)+(R6) is a pullback. Hence, by the Injectivity Lemma A.14, the injectivity of $R_1 \to D_2 \to M$ carries over to $K_1 \to D_0 \to \bar{D}_1$ and $L_1 \to \bar{D}_2 \to \bar{H}$. This completes the proof. □

Next we combine the Sequential Commutativity Theorem I with the Simulation Theorem.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

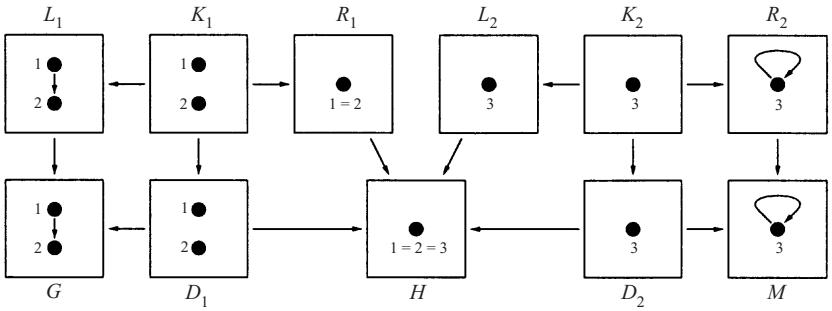


Fig. 13. Ambiguity in sequential independence

Theorem 6.8. (Sequential commutativity III) In $\mathrm{DPO}^{a/i}$, 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 $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{Q(p_1)} M$.

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

To see that the resulting direct derivations in Theorem 6.8 need not be sequentially independent, consider the sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ of Example 6.5. There the resulting direct derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{Q(p_1)} M$, which are depicted in Figure 9, are not sequentially independent.

We conclude this section by showing that in the presence of non-injective rules, interchanging two sequentially independent steps $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ may result in different derivations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ and $G \Rightarrow_{p_2} \bar{H}' \Rightarrow_{p_1} M$ such that $\bar{H}$ and $\bar{H}'$ are not isomorphic.

Example 6.9. The direct derivations in Figure 13 are sequentially independent: there exist two graph morphisms from L_2 to D_1 and a unique graph morphism from R_1 to D_2 satisfying the conditions of Definition 6.1. Depending on which of the two morphisms from L_2 to D_1 is chosen, the construction in the proof of the Sequential Commutativity Theorem I yields the first or the second derivation in Figure 14. Note that the middle graphs of the two derivations are not isomorphic.

By Theorem 6.4 (6.7), we have that two (strongly) sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ can be interchanged. The result of the interchange depends on the chosen independence pair.

Definition 6.10. (Interchange operator IC**)** Let $\Delta\colon G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be two (strongly) sequentially independent direct derivations with independence pair π as in Definition 6.1. Then $\mathrm{IC}_\pi(\Delta)$ denotes the two (strongly) sequentially independent direct derivations

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

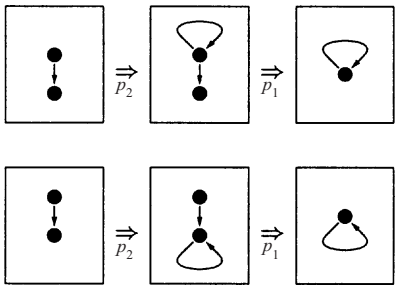


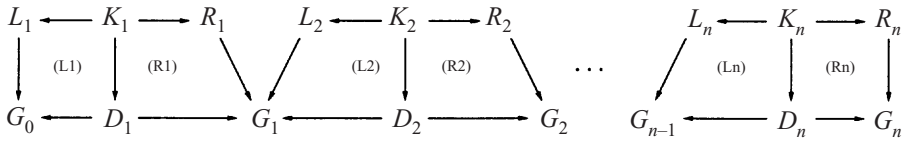
Fig. 14. Interchanging the steps of Figure 13 in different ways

$G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ constructed in the proof of Theorem 6.4 (6.7) and $\bar{\pi}$ denotes the associated independence pair $\langle R_2 \to \bar{D}_1, L_1 \to \bar{D}_2 \rangle$.

The result of an interchange are two (strongly) sequentially independent direct derivations, so these direct derivations can again be interchanged. The following result shows that by choosing the induced independence pair for this, the interchange results in the original direct derivations.

Definition 6.11. (Generalized instance, isomorphic derivation)

1 Let $\Delta : G_0 \Rightarrow_{p_1} G_1 \Rightarrow_{p_2} \ldots \Rightarrow_{p_n} G_n$ be a derivation of the form



and (L1′), (R1′), …, (Ln′), (Rn′) as in Figure 15 be pushouts. Then the derivation $\Delta' : G'_0 \Rightarrow_{p_1} G'_1 \Rightarrow_{p_2} \ldots \Rightarrow_{p_n} G'_n$ consisting of the composed pushouts (L1)+(L1′), …, (Rn)+(Rn′) is a *generalized instance* of Δ. We say that Δ' is *based* on the morphism $G_0 \to G'_0$.

2 A generalized instance of Δ based on an injective morphism is said to be an *instance* (Plump 1995).

3 Two derivations Δ and Δ' are *isomorphic*, denoted by $\Delta \cong \Delta'$, if Δ' is an instance of Δ based on an isomorphism.

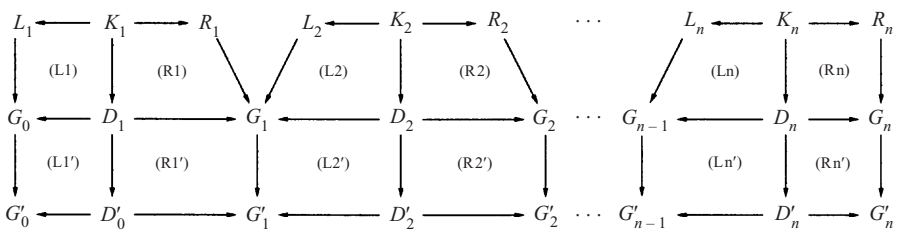


Fig. 15. A generalized instance of the derivation in Definition 6.11

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Note that for isomorphic derivations Δ and Δ', all vertical morphisms relating the derivations are isomorphisms (Injectivity and Surjectivity Lemma A.14 and A.15).

Theorem 6.12. (Interchange) Let $\Delta: G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be two direct derivations that are (strongly) sequentially independent by an independence pair π. Then $\mathrm{IC}_{\bar{\pi}}(\mathrm{IC}_{\pi}(\Delta)) \cong \Delta$.

Proof. Let $\mathrm{IC}_{\pi}(\Delta)$ and $\bar{\pi}$ be constructed as in the proof of Theorem 6.4 (6.7). Then all diagrams in Figure 11 are pushouts. Moreover, by the Pushout–Pullback Lemma A.11, diagram (8) is a pullback as well. As a consequence, the decomposition of the diagrams of $\mathrm{IC}_{\pi}(\Delta)$ according to $\bar{\pi}$ yields the diagrams depicted in Figure 11. Since diagram (7) in Figure 10 is a pushout as well, a rearrangement of the diagrams in Figure 11 and a construction of a pushout object of $D_1 \leftarrow D_0 \rightarrow D_2$ yields diagrams as depicted in Figure 10. Therefore, the derivations $\mathrm{IC}_{\bar{\pi}}(\mathrm{IC}_{\pi}(\Delta))$ and Δ are equal up to isomorphism. □

In $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$, for every pair of sequentially independent direct derivations there is a unique independence pair. As a consequence, choosing any independence pair of $\mathrm{IC}_{\pi}(\Delta)$ for a second interchange will give back the original derivation. A corresponding result does not hold in $\mathrm{DPO}^{\mathrm{a/i}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$ where choosing an arbitrary independence pair of $\mathrm{IC}_{\pi}(\Delta)$ may give a derivation that is not isomorphic to the original one.

Example 6.13. Consider the first derivation $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ in Figure 14. The direct derivations are sequentially independent and there is a unique independence pair π. The result of the interchange is the derivation $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in Figure 13. As discussed in Example 6.9, there are two independence pairs $\bar{\pi}_1$ and $\bar{\pi}_2$. Depending on which independence pair is chosen, the second interchange yields the first or the second derivation in Figure 14. While the first derivation coincides with the original derivation, the second derivation is not isomorphic to the original one.

7. Parallelism

In this section we consider sequentially independent direct derivations, parallel rules, and derivations through parallel rules. The connection between sequentially independent direct derivations and parallel derivations is established in the Parallelism Theorem.

We now define the parallel composition of rules p_1 and p_2 as the disjoint union of p_1 and p_2, that is, the disjoint union of the left-hand sides, the right-hand sides, and the interfaces.

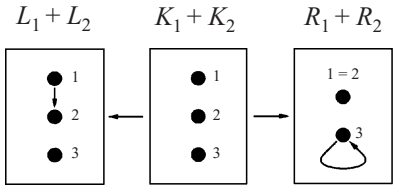
The *disjoint union* of two graphs G_1 and G_2 is the graph $G_1 + G_2$ with node set $(V_{G_1} \times \{1\}) \cup (V_{G_2} \times \{2\})$ and edge set $(E_{G_1} \times \{1\}) \cup (E_{G_2} \times \{2\})$. An edge $\langle e, i\rangle$ with $e \in E_{G_i}$, $i \in \{1,2\}$, has the source $\langle \mathrm{s}_{G_i}(e), i\rangle$, the target $\langle \mathrm{t}_{G_i}(e), i\rangle$, and the label $\mathrm{m}_{G_i}(e)$. A node $\langle v, i\rangle$ has the label $\mathrm{l}_{G_i}(e)$. The *disjoint union* of two graph morphisms $g_1: G_1 \rightarrow H_1$ and $g_2: G_2 \rightarrow H_2$ is the morphism $g_1 + g_2: G_1 + G_2 \rightarrow H_1 + H_2$ consisting of the functions $g_{1_V} + g_{2_V}$ and $g_{1_E} + g_{2_E}$ mapping a node $\langle v, i\rangle$ to $\langle g_{i_V}(v), i\rangle$ and an edge $\langle e, i\rangle$ to $\langle g_{i_E}(e), i\rangle$.

Definition 7.1. (Parallel rules and derivations) The *parallel rule* $p_1 + p_2$ of p_1 and p_2 is defined by $\langle L_1 + L_2 \leftarrow K_1 + K_2 \rightarrow^{r_1 + r_2} R_1 + R_2\rangle$, where $+$ denotes the disjoint union of

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

graphs, respectively, graph morphisms. A direct derivation $G \Rightarrow_{p_1+p_2} M$ is called a *parallel direct derivation.*

Example 7.2. Consider the rules p_1 and p_2 in Figure 13. Then the parallel rule $p_1 + p_2$ of p_1 and p_2 is given below. It can be applied to the graph G in Figure 13 yielding a parallel direct derivation $G \Rightarrow_{p_1+p_2} M$ in $\mathrm{DPO}^{a/a}$.



A connection between sequentially independent direct derivations and parallel direct derivations is expressed by the following (known) result.

Theorem 7.3. (Parallelism I) In $\mathrm{DPO}^{i/a}$ and $\mathrm{DPO}^{a/a}$ the following hold:

(1) For every pair of sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ there is a parallel derivation $G \Rightarrow_{p_1+p_2} M$.
(2) For every parallel derivation $G \Rightarrow_{p_1+p_2} M$ there are sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ and $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$.

The statement is proved in Ehrig and Kreowski (1976) and Kreowski (1977). Since the proof has to be inspected a second time, it is given below.

Proof.
(1) Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be sequentially independent. Then the derivation can be decomposed as in the proof of Theorem 6.4 and all diagrams in Figure 10 are pushouts. The pushouts can be composed as butterflies (see Figure 16). By the Butterfly Lemma A.26, the diagrams (L1) and (R1) in Figure 17 are pushouts. As a consequence, there is a parallel derivation $G \Rightarrow_{p_1+p_2} M$.
(2) Let $G \Rightarrow_{p_1+p_2} M$ be a parallel derivation. Then we have folded butterflies as in Figure 17. By the Butterfly Lemma A.26, there exists a decomposition of the pushouts (L1) and (R1) into pushouts (L3), (L4), (L5), and (R3), (R4), (R6) (see Figure 16). Composing the diagrams in a new fashion and constructing H as the pushout object of $D_1 \leftarrow D_0 \rightarrow D_2$ leads to Figure 10. By the Composition Lemma A.18, the composed diagrams are pushouts. Moreover, $R_1 \rightarrow H = R_1 \rightarrow D_2 \rightarrow H$ and $L_2 \rightarrow H = L_2 \rightarrow D_1 \rightarrow H$, that is, the two direct derivations $G \Rightarrow_{r_1} H \Rightarrow_{r_2} M$ are sequentially independent. Accordingly, one may construct sequentially independent direct derivations $G \Rightarrow_{r_2} \bar{H} \Rightarrow_{r_1} M$. □

The following counter-example demonstrates that the parallelism property does not hold in $\mathrm{DPO}^{i/i}$ and $\mathrm{DPO}^{a/i}$.

Example 7.4. The direct derivations $L \Rightarrow_p L \Rightarrow_p L$ through $p = \langle L \leftarrow L \rightarrow L \rangle$, where L is some non-empty graph, are sequentially independent in $\mathrm{DPO}^{i/i}$. However, there does not exist a direct derivation $L \Rightarrow_{p+p} L$ in $\mathrm{DPO}^{i/i}$.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

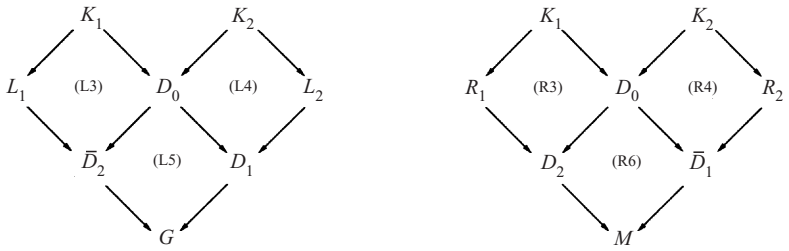


Fig. 16. Butterflies

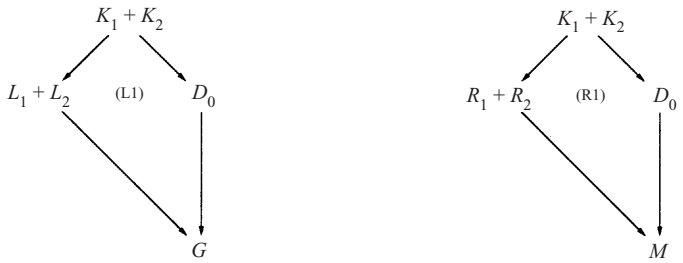
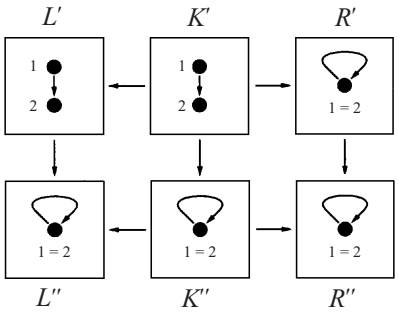


Fig. 17. Folded butterflies

The counter-example suggests that we consider quotient rules of parallel rules. But, in $\mathrm{DPO}^{a/i}$ and $\mathrm{DPO}^{i/i}$, it is not the case that for every direct derivation through a quotient rule of $p_1 + p_2$ there are also sequentially independent direct derivations through p_1 and p_2.

Example 7.5. Consider the rules p_1 and p_2 in Figure 12. Then the rules $p' = \langle L' \leftarrow K' \rightarrow R' \rangle$ and $p'' = \langle L'' \leftarrow K'' \rightarrow R'' \rangle$ below are quotient rules of $p_1 + p_2$.



The rule p'' may be applied to the graph M in Figure 12 yielding a direct derivation $M \Rightarrow_{p''} M$ in $\mathrm{DPO}^{a/i}$. But there are no direct derivations $M \Rightarrow_{p_1} M$ and $M \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{a/i}$. The reason is that the morphisms $L_1 \rightarrow L_1 + L_2 \rightarrow L''$ and $L_2 \rightarrow L_1 + L_2 \rightarrow L''$ are not injective. (If we consider injective rules instead of arbitrary ones, we get a similar situation.) For the rule p', the morphisms $L_1 \rightarrow L_1 + L_2 \rightarrow L'$ and $L_2 \rightarrow L_1 + L_2 \rightarrow L'$ are injective. It may be applied to the graph G in Figure 12 yielding a direct derivation $G \Rightarrow_{p'} M$ in $\mathrm{DPO}^{a/i}$. Moreover, there is a direct derivation $G \Rightarrow_{p_2} M$. But there is no

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

direct derivation $M \Rightarrow_{p_1} M$ in DPO$^{a/i}$. The reason is that the morphism $K' \to R'$ identifies nodes that have preimages in K_1 as well as K_2.

For DPO$^{a/i}$ and DPO$^{i/i}$, we have to consider the following subclass of quotient rules of parallel rules.

Definition 7.6. (Proper quotient rules) A rule $p' = \langle L' \leftarrow K' \rightarrow R' \rangle$ is a *proper quotient rule* of a parallel rule $p_1 + p_2$ if:

(a) $p' \in Q(p_1 + p_2)$,
(b) $L_i \to L_1 + L_2 \to L'$ is injective for $i = 1, 2$, and
(c) $S \to K' \to R'$ is injective, where S and $S \to K'$ are constructed as follows: $K_i \to K' = K_i \to K_1 + K_2 \to K'$, S is a pullback object of $K_1 \to K' \leftarrow K_2$ and $S \to K' = S \to K_1 \to K'$.

The set of proper quotient rules of $p_1 + p_2$ is denoted by $\mathrm{PQ}(p_1 + p_2)$.

For injective rules p_1 and p_2, a rule p' is already a proper quotient rule of $p_1 + p_2$ if the conditions (a) and (b) are satisfied: the injectivity of $L_i \to L_1 + L_2 \to L'$ implies the injectivity of $K_i \to K_1 + K_2 \to K'$ $(i = 1, 2)$. By the Injectivity Lemma A.14, $S \to K_i$ for $i = 1, 2$ is injective, too. By the injectivity of $K_i \to R_i$ for $i = 1, 2$, $K_1 + K_2 \to R_1 + R_2$ and $K' \to R'$ are injective. Therefore, $S \to K' \to R'$ is injective.

Note that in Section 9 a simple method for constructing the set of all proper quotient rules of $p_1 + p_2$ is presented. It is based on the idea of 'gluing' the left-hand sides, the right-hand sides, and the interfaces of p_1 and p_2 according to a graph S that relates the rules p_1 and p_2 by graph morphisms $S \to K_i$ such that $S \to K_i \to R_i$ is injective.

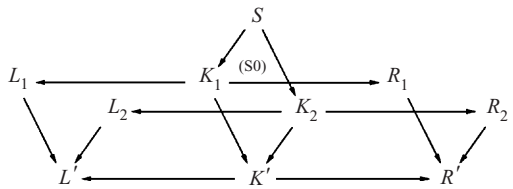
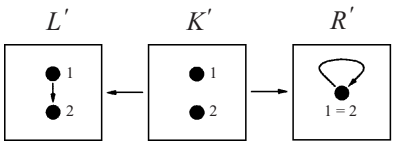


Fig. 18. Illustration for Definition 7.6

Example 7.7. Consider the rules p_1 and p_2 in Figure 13. The rule p' below is a proper quotient rule of $p_1 + p_2$. It can be applied to the graph G in Figure 13 yielding a parallel direct derivation $G \Rightarrow_{p'} M$ in DPO$^{a/i}$.



Theorem 7.8. (Parallelism II) In DPO$^{a/i}$ and DPO$^{i/i}$ the following hold:

(1) For every pair of strongly sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ there is a direct derivation $G \Rightarrow_{\mathrm{PQ}(p_1+p_2)} M$.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

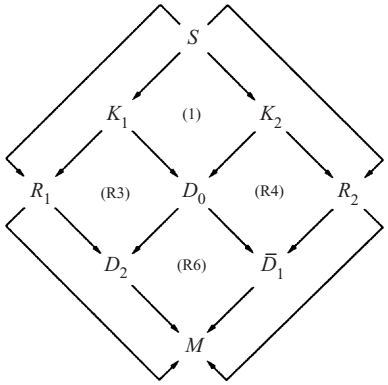


Fig. 19. Illustration for the proof of Theorem 7.8

(2) For every direct derivation $G \Rightarrow_{\mathrm{PQ}(p_1+p_2)} M$ there are strongly sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ and $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$.

In the case of $\mathrm{DPO}^{\mathrm{i/i}}$, Theorem 7.8 may be simplified by replacing strong sequential independence by sequential independence.

Proof.
(1) Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$ be strongly sequentially independent. Then the direct derivations are in $\mathrm{DPO}^{\mathrm{a/a}}$, and, by Theorem 7.3, there is a parallel derivation $G \Rightarrow_{p_1+p_2} M$ in $\mathrm{DPO}^{\mathrm{a/a}}$. By the Simulation Theorem, there is a direct derivation $G \Rightarrow_{p'} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$ for some $p' \in \mathrm{Q}(p_1 + p_2)$. It remains to show that $p' \in \mathrm{PQ}(p_1 + p_2)$. Consider Figure 8. By assumption, $L_1 \to G$ and $L_2 \to H$ are injective. Moreover, $L_2 \to D_1$ and $D_1 \to G$ are injective. Therefore, $L_2 \to G = L_2 \to D_1 \to G$ is injective. Since the morphism $L_i \to L_1 + L_2 \to L' \to G = L_i \to L_1 + L_2 \to G = L_i \to G$ for $i = 1, 2$ is injective, $L_i \to L_1 + L_2 \to L'$ for $i = 1, 2$ is injective. As a consequence, $K_i \to K_1 + K_2 \to K'$ for $i = 1, 2$ is injective. Since (S0) in Figure 18 is a pullback, $S \to K_i$ for $i = 1, 2$ is injective. Since $S \to K_2$ and $K_2 \to D_2 \to D_0 \to D_2$ are injective and (1)+(R3) commutes, $S \to K_1 \to R_1$ is injective (see Figure 19). By strong sequential independence, $R_1 \to D_2 \to M$ is injective. Therefore, $S \to K' \to R' \to M$ and $S \to K' \to R'$ are injective.

(2) Let $G \Rightarrow_{p'} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$ for some $p' \in \mathrm{PQ}(p_1 + p_2)$. By the Simulation Theorem, there is a parallel derivation $G \Rightarrow_{p_1+p_2} M$ in $\mathrm{DPO}^{\mathrm{a/a}}$, and, by Theorem 7.3, there are sequentially independent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ and $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ in $\mathrm{DPO}^{\mathrm{a/a}}$. It remains to show that the direct derivations are in $\mathrm{DPO}^{\mathrm{a/i}}$ and are strongly sequentially independent. Since $p' \in \mathrm{PQ}(p_1 + p_2)$, we have $L_i \to L_1 + L_2 \to L'$ for $i = 1, 2$ and $S \to K' \to R'$ are injective. Moreover, by assumption, $L' \to G$ is injective. Therefore, the morphism $L_i \to G$ is injective, for $i = 1, 2$. Since diagram (S0) in Figure 18 is a pullback and the morphism $K' \to D_0$ is injective, diagram (1) in Figure 19 is a pullback (Pullback Inheritance A.16). Now the injectivity of $S \to K' \to R'$ implies the injectivity of $S \to K_i \to R_i$, for $i = 1, 2$. This implies the injectivity of $K_2 \to D_0 \to D_2$ and $K_1 \to D_0 \to \bar{D}_1$ (Lifting of injectivity A.25), see Figure 19. By the Injectivity Lemma A.14,

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

the morphisms $L_2 \to D_1 \to H$ and $L_1 \to \bar{D}_2 \to \bar{H}$ are injective (see Figure 10 and 11). Therefore, the direct derivations are in $\mathrm{DPO}^{\mathrm{a/i}}$. By the same argument, the morphisms $R_2 \to \bar{D}_1 \to M$ and $R_1 \to D_2 \to M$ are injective, so the direct derivations are strongly sequentially independent. □

Note that Theorem 7.8 does not hold for $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$: since strong sequential independence implies sequential independence and every parallel rule is proper, the first statement of Theorem 7.8 holds for $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$. Inspecting the proof of the second statement, one may see that the sequentially independent direct derivations need not be strongly sequentially independent; thus the second statement does not hold.

We conclude this section, by showing that in the presence of non-injective rules, the parallelization of two sequentially independent steps $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ may result in direct derivations $G \Rightarrow_{p_1+p_2,g} M$ and $G \Rightarrow_{p_1+p_2,g'} M$ such that g and g' are different, and the sequentialization of a direct derivation $G \Rightarrow_{\mathrm{PQ}(p_1+p_2)} M$ may result in different derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ and $G \Rightarrow_{p_1} H' \Rightarrow_{p_2} M$ such that H and H' are not isomorphic.

Example 7.9. The direct derivations in Example 6.9 are sequentially independent. But there exist two distinct graph morphisms $L_2 \to D_1$ with $L_2 \to D_1 \to H = L_2 \to H$. As a consequence, there are two distinct parallel derivations $G \Rightarrow_{p_1+p_2} M$. Moreover, there exist two distinct graph morphisms $L_1 + L_2 \to L'$ for constructing the quotient rule p' of $p_1 + p_2$ in Example 7.7. As a consequence, there are two distinct sequentializations $G \Rightarrow_{p_2} \bar{H} \Rightarrow_{p_1} M$ and $G \Rightarrow_{p_2} \bar{H}' \Rightarrow_{p_1} M$ of the direct derivation $G \Rightarrow_{p'} M$.

8. Concurrency

In this section, we consider again derivations of the form $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$, but the assumption of sequential independence is dropped. This leads to the notion of a dependence relation s for a pair of rules, sequentially s-dependent derivations, and the construction of s-concurrent rules and derivations. The connection between sequentially s-dependent derivations and s-concurrent derivations is established in the so-called Concurrency Theorem.

By the parallelism theorems presented in Section 7, we know that two sequentially independent derivation steps $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ may be replaced by a single parallel derivation step $G \Rightarrow_{p_1+p_2} M$. It remains the question whether we can do something similar if the two steps $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ are not sequentially independent, that is, if the rule p_2 needs some items that are generated by p_1 or if p_2 deletes some items that are needed by p_1. In the following, we determine the dependencies between the right-hand side of p_1 and the left-hand side of p_2, construct the sequential composition of p_1 and p_2 with respect to the dependence relation s (the so-called s-concurrent rule), and show that two sequentially s-dependent derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ may be replaced by a single derivation step $G \Rightarrow M$ through an s-concurrent rule.

Example 8.1. Figure 20 shows two sequentially dependent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{\mathrm{a/i}}$ and their canonical dependence relation $s = \langle R_1 \leftarrow S \rightarrow L_2 \rangle$. The latter is constructed as the pullback of the morphisms $R_1 \to H \leftarrow L_2$.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

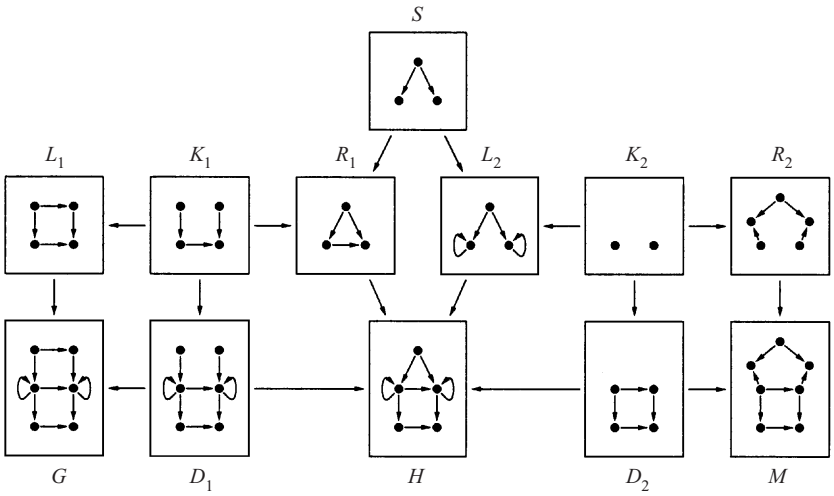


Fig. 20. Two sequentially dependent steps and their canonical dependence relation

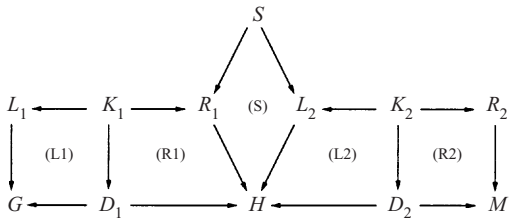


Fig. 21. A derivation

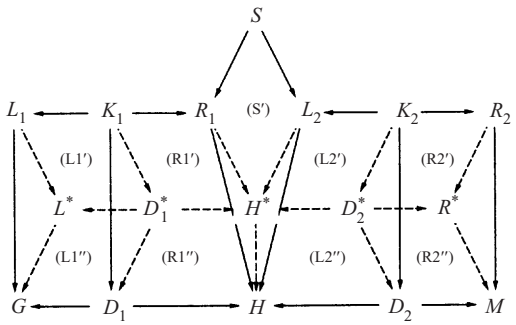


Fig. 22. Decomposition of the derivation in Figure 21

Lemma 8.2. (Construction of the canonical dependence relation) In DPO$^{i/a}$, DPO$^{a/i}$ and DPO$^{i/i}$, for every derivation $\Delta : G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ as in Figure 21 there is a pair of morphisms $s = \langle R_1 \leftarrow S \rightarrow L_2 \rangle$ and a derivation $\Delta' : L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ as in Figure 22 such that Δ is a generalized instance of Δ', (S) is a pullback and (S′) is a pushout.

Proof. Let $\Delta : G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be a derivation as shown in Figure 21. Construct S as

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

the pullback object of $R_1 \to H \leftarrow L_2$ and H^* as the pushout object of $R_1 \leftarrow S \to L_2$. Then diagram (S) in Figure 21 is a pullback and diagram (S′) in Figure 22 is a pushout. Moreover, there exists a unique graph morphism $H^* \to H$ such that $R_1 \to H^* \to H = R_1 \to H$ and $L_2 \to H^* \to H = L_2 \to H$. Now construct D_i^* as pullback object of $D_i \to H \leftarrow H^*$ $(i = 1, 2)$. Then there exist unique graph morphisms $K_i \to D_i^*$ such that $K_i \to D_i^* \to D_i = K_i \to D_i$ and the diagrams (R1′) and (L2′) in Figure 22 commute. Finally, construct L^* and R^* as pushout objects of $L_1 \leftarrow K_1 \to D_1^*$ and $R_2 \leftarrow K_2 \to D_2^*$, respectively. Then there are unique graph morphisms $L^* \to G$ and $R^* \to M$, such that $L_1 \to L^* \to G = L_1 \to G$, $R_2 \to R^* \to M = R_2 \to M$, and the diagrams (L1″) and (R2″) commute. Moreover, all diagrams in Figure 22 are pushouts.

This may be seen as follows. Since $K_2 \to L_2$ is injective, by the Twisted-Triple-Pushout Lemma A.22, (L2′) and (L2″) are pushouts. In DPO$^{\mathrm{i/a}}$ and DPO$^{\mathrm{i/i}}$, $K_1 \to R_1$ is injective, as well, and by the Twisted-Triple-Pushout Lemma A.22, (R1′) and (R1″) are pushouts. In DPO$^{\mathrm{a/i}}$, $R_1 \to H$ and $L_2 \to H$ are injective and, by the Inheritance of Injectivity A.17 and the Twisted-Triple-Pushout Lemma A.22, $H^* \to H$ is injective and (R1′) and (R1″) are pushouts. Moreover, (L1′) and (R2′) are pushouts and, by the Decomposition Lemma A.19, (L1″) and (R2″) are pushouts.

Consequently, $\Delta' : L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ is a derivation and Δ is a generalized instance of Δ'. □

Note that Lemma 8.2 does not hold for DPO$^{\mathrm{a/a}}$. But given a derivation $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in DPO$^{\mathrm{a/a}}$, we may switch to the corresponding derivation $G \Rightarrow_{\mathrm{Q}(p_1)} H \Rightarrow_{\mathrm{Q}(p_2)} M$ in DPO$^{\mathrm{a/i}}$ and construct a canonical dependence relation for it.

Lemma 8.2 motivates us to define a dependence relation as follows.

Definition 8.3. (Dependence relation and sequential *s*-dependence)

1 A *dependence relation* $s = \langle R_1 \leftarrow S \to L_2 \rangle$ for $\langle p_1, p_2 \rangle$ is a pair of graph morphisms such that there exists a derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ as in Figure 22 for which diagram (S′) is a pushout. Such a derivation is said to be an *s-derivation* for $\langle p_1, p_2 \rangle$.

2 Given a dependence relation s for $\langle p_1, p_2 \rangle$, two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ as in Figure 21 are *sequentially s-dependent* if diagram (S) commutes and $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ for $\langle p_1, p_2 \rangle$ as in Figure 22, where $H^* \to H$ is the unique morphism that exists by the universal property of the pushout (S′).

Note that $\epsilon = \langle R_1 \leftarrow \varnothing \to L_2 \rangle$ is a dependence relation for $\langle p_1, p_2 \rangle$, called the *empty* dependence relation. (Here $\varnothing$ denotes the empty graph.) $L_1 + L_2 \Rightarrow_{p_1} R_1 + L_2 \Rightarrow_{p_2} R_1 + R_2$ is an ϵ-derivation and every pair of sequentially independent direct derivations is sequentially ϵ-dependent.

Example 8.4. The pair $s = \langle R_1 \leftarrow S \to L_2 \rangle$ of morphisms in Figure 20 forms a dependence relation for $\langle p_1, p_2 \rangle$. Moreover, the two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ are sequentially s-dependent.

The following example demonstrates that not every pair of morphisms defines a dependence relation.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

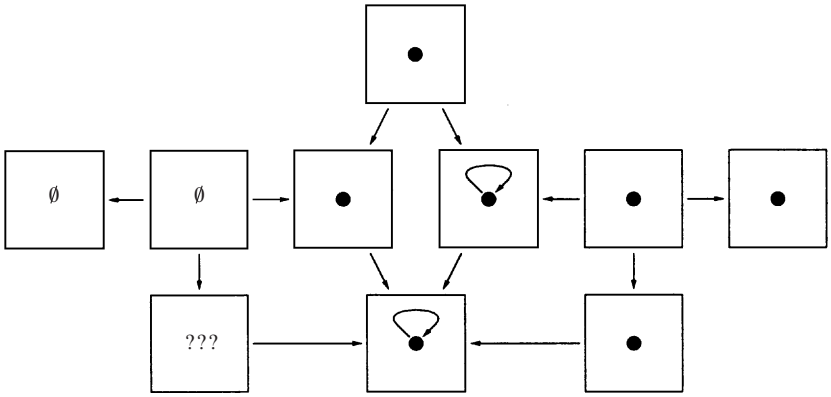


Fig. 23. A counter-example for a dependence relation

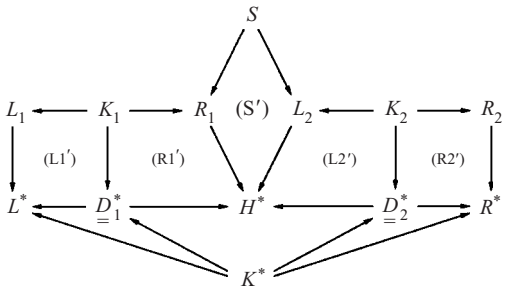


Fig. 24. An s-concurrent rule of $\langle p_1, p_2 \rangle$

Example 8.5. Consider the pair $s = \langle R_1 \leftarrow S \rightarrow L_2 \rangle$ of morphisms in Figure 23. There does not exist a derivation $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ reflecting the dependencies described by s. In particular, there does not exist an s-derivation for $\langle p_1, p_2 \rangle$.

Given a dependence relation s for $\langle p_1, p_2 \rangle$, we may construct an s-concurrent rule for $\langle p_1, p_2 \rangle$ from an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ by choosing L^* and R^* as the left- and the right-hand side of the rule, respectively, and constructing the interface K^* of the rule as pullback object of the intermediate graphs D_1 and D_2.

Definition 8.6. (s-concurrent rules and derivations) Given a dependence relation s for $\langle p_1, p_2 \rangle$, an *s-concurrent rule* $p^* = \langle L^* \leftarrow K^* \rightarrow R^* \rangle$ of $\langle p_1, p_2 \rangle$ is constructed from an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ as in Figure 24 by constructing the interface K^* as a pullback object of $D_1^* \rightarrow H^* \leftarrow D_2^*$ and defining $K^* \rightarrow L^* = K^* \rightarrow D_1^* \rightarrow L^*$ and $K^* \rightarrow R^* = K^* \rightarrow D_2^* \rightarrow R^*$. The set of all s-concurrent rules of $\langle p_1, p_2 \rangle$ is denoted by $\mathrm{Con}(p_1, s, p_2)$.

A direct derivation through an s-concurrent rule is called an *s-concurrent direct derivation.*

Since the number of non-isomorphic pushout complements of $K_1 \rightarrow R_1 \rightarrow H^*$ is finite, we can without loss of generality assume that $\mathrm{Con}(p_1, s, p_2)$ is finite. If the rule p_1 is

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

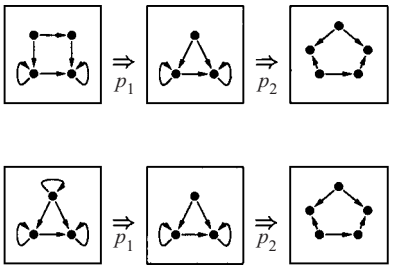


Fig. 25. Two s-derivations

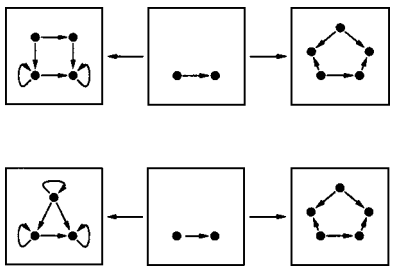


Fig. 26. Two s-concurrent rules

injective, there exists a unique pushout complement of $K_1 \to R_1 \to H^*$ and a unique s-concurrent rule of $\langle p_1, p_2 \rangle$.

Note that every parallel rule $p_1 + p_2$ is an ϵ-concurrent rule of $\langle p_1, p_2 \rangle$ and every parallel derivation is an ϵ-concurrent derivation.

Example 8.7. Consider the dependence relation s for $\langle p_1, p_2 \rangle$ in Figure 20. There are two non-isomorphic s-derivations, which are shown in Figure 25. Depending on which of the two derivations is chosen, the construction of an s-concurrent rule yields either the first or the second rule in Figure 26. Applying the first rule to the graph G of Figure 25 yields an s-concurrent derivation $G \Rightarrow M$.

A connection between sequentially s-dependent direct derivations and s-concurrent derivations is expressed by the following result. Up to now, the result was only known for $\mathrm{DPO}^{\mathrm{i/a}}$ (Ehrig *et al.* 1986; Ehrig *et al.* 1991).

Theorem 8.8. (Concurrency I) In $\mathrm{DPO}^{\mathrm{i/a}}$ and $\mathrm{DPO}^{\mathrm{a/a}}$ the following hold:

(1) For every pair of sequentially s-dependent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ there is an s-concurrent derivation $G \Rightarrow_{\mathrm{Con}(p_1,s,p_2)} M$.
(2) For every s-concurrent derivation $G \Rightarrow_{\mathrm{Con}(p_1,s,p_2)} M$ there are two sequentially s-dependent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$.

Proof.
(1) Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be sequentially s-dependent. Then the derivation is a generalized instance of an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ (see Figure 27). Without loss of generality, we may assume that the pair $\langle D_1^* \to H^*, D_1^* \to D_1 \rangle$ is monomorphic (see Definition A.6).

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

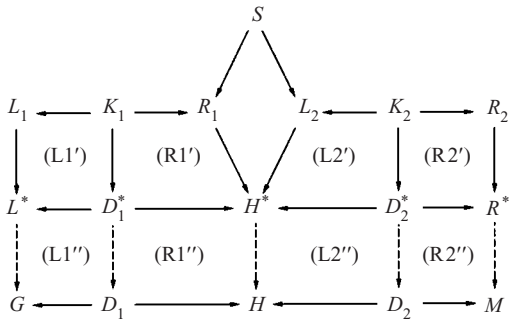


Fig. 27. A generalized instance of an *s*-derivation for $\langle p_1, p_2 \rangle$

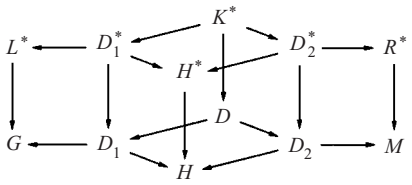


Fig. 28. Synthesis

This may be seen as follows. In $\mathrm{DPO}^{\mathrm{i/a}}$, $D_1^* \to H^*$ is injective because $K_1 \to R_1$ is. In $\mathrm{DPO}^{\mathrm{a/a}}$, the pair $\langle D_1^* \to H^*, D_1^* \to D_1 \rangle$ need not be monomorphic. By the Monomorphic-Pair Lemma A.24, there is a factorization of (R1′)+(R1″) into pushouts $K_1 R_1 \bar{D}_1^* H^*$ and $\bar{D}_1^* H^* D_1 H$ such that $\langle \bar{D}_1^* \to H^*, \bar{D}_1^* \to D_1 \rangle$ is monomorphic. Moreover, there is a factorization of (L1′)+(L1″) into pushouts $K_1 L_1 \bar{D}_1^* \bar{L}^*$ and $\bar{D}_1^* \bar{L}^* D_1 G$. Therefore, $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of the *s*-derivation $\bar{L}^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$.

Now let p^* be constructed from this *s*-derivation. Then K^* is a pullback object of $D_1^* \to H^* \leftarrow D_2^*$ (see Figure 28). Construct D as a pullback object of $D_1 \to H \leftarrow D_2$. Then there exists a unique morphism $K^* \to D$ such that $K^* \to D \to D_i = K^* \to D_i^* \to D_i$ $(i = 1, 2)$. By the Cube Pullback–Pushout Lemma A.13, the diagrams $K^* D D_1^* D_1$ and $K^* D D_2^* D_2$ are pushouts and, by Composition Lemma A.18, the composed diagrams $K^* D L^* G$ and $K^* D R^* M$ are pushouts. Thus $G \Rightarrow_{p^*} M$.

(2) Let $G \Rightarrow_{p^*} M$ be an *s*-concurrent derivation. Then p^* is constructed from an *s*-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ (see Figure 29).

In the following, we will construct diagrams as depicted in Figure 27. To this end, construct D_i as pushout object of $D_i^* \leftarrow K^* \to D$ $(i = 1, 2)$. Then there exist unique graph morphisms $D_1 \to G$ and $D_2 \to M$ such that the diagrams (L1″) and (R2″) commute. By the Decomposition Lemma A.19 the new diagrams (L1″) and (R2″) are pushouts. Now construct H as pushout object of $D_1 \leftarrow D_1^* \to H^*$ (see diagram (R1″)). Then there exists a unique graph morphism $D_2 \to H$ such that (L2″) commutes. By the Decomposition Lemma A.19, (L2″) is a pushout as well. Now all diagrams in Figure 27 are pushouts. By the Composition Lemma A.18, the composed diagrams are pushouts. Therefore,

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

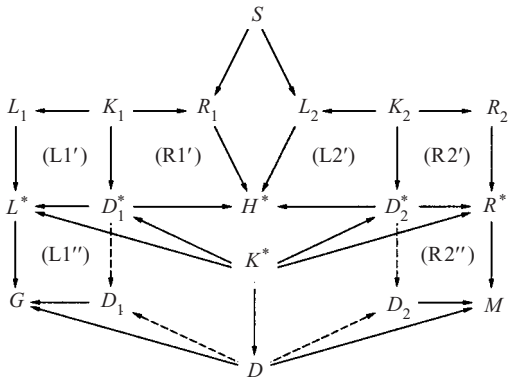


Fig. 29. Analysis

$G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$, and hence the direct derivations are sequentially s-dependent. □

The following counter-example demonstrates that the concurrency property does not hold in DPO$^{i/i}$ and DPO$^{a/i}$.

Example 8.9. The direct derivations $L \Rightarrow_p L \Rightarrow_p L$ in Example 7.4 are sequentially ϵ-dependent in DPO$^{i/i}$ and $p + p$ is the unique ϵ-concurrent rule of $\langle p, p \rangle$. But there does not exist a direct derivation $L \Rightarrow_{p+p} L$ in DPO$^{i/i}$. For the dependence relation $s = \langle L \leftarrow L \rightarrow L \rangle$, where $L \rightarrow L$ is the identity, p is the unique s-concurrent rule $\langle p, p \rangle$ and $L \Rightarrow_p L$ is a derivation in DPO$^{i/i}$.

We strengthen sequential s-dependence by requiring that the derivation is an instance of an s-derivation, that is, the vertical morphisms relating the s-derivation to the original derivation are injective. Then the constructed s-concurrent rule can be applied in an injective manner.

Definition 8.10. (Strong sequential s-dependence) Given a dependence relation s for $\langle p_1, p_2 \rangle$, two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ as in Figure 21 are *strongly sequentially s-dependent* if diagram (S) commutes and $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is an instance of an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ for $\langle p_1, p_2 \rangle$, where $H^* \rightarrow H$ is the unique morphism that exists by the universal property of the pushout (S′) in Figure 22.

In DPO$^{a/i}$ and DPO$^{i/i}$, it is not the case that for every direct s-concurrent derivation there are also sequentially s-dependent direct derivations. If the used s-concurrent rule is constructed from an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$, we have to ensure that the vertical morphisms $L_1 \rightarrow L^*$ and $L_2 \rightarrow H^*$ are injective. Hence we restrict on s-concurrent rules constructed from s-derivations in DPO$^{a/i}$.

Definition 8.11. (Proper s-concurrent rules) An s-concurrent rule for $\langle p_1, p_2 \rangle$ is *proper* if it is constructed from an s-derivation for $\langle p_1, p_2 \rangle$ in DPO$^{a/i}$. The set of all proper s-concurrent rules of $\langle p_1, p_2 \rangle$ is denoted by $\mathrm{PCon}(p_1, s, p_2)$.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Example 8.12. The first rule in Figure 26 is a proper s-concurrent rule for $\langle p_1, p_2 \rangle$ in Figure 20. The second rule $p^* = \langle L^* \leftarrow K^* \rightarrow R^* \rangle$ is constructed from an s-derivation not in $\mathrm{DPO}^{a/i}$ and thus is a not proper. Note that there is an s-concurrent derivation $L^* \Rightarrow_{p^*} R^*$ in $\mathrm{DPO}^{a/i}$, but there are not two sequentially s-dependent direct derivations $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ in $\mathrm{DPO}^{a/i}$.

Theorem 8.13. (Concurrency II) In $\mathrm{DPO}^{a/i}$ and $\mathrm{DPO}^{i/i}$ the following hold:

(1) For every pair of strongly sequentially s-dependent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ there is an s-concurrent derivation $G \Rightarrow_{\mathrm{PCon}(p_1,s,p_2)} M$.
(2) For every s-concurrent derivation $G \Rightarrow_{\mathrm{PCon}(p_1,s,p_2)} M$ there are strongly sequentially s-dependent direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$.

Proof. The statement follows by inspecting the proof of Theorem 8.8. Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{a/i}$ be strongly sequentially s-dependent. Then the derivation is an instance of an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ in $\mathrm{DPO}^{a/i}$. Let p^* be an s-concurrent rule constructed from this s-derivation. Then p^* is proper and $G \Rightarrow_{p^*} M$ is in $\mathrm{DPO}^{a/i}$. For the opposite direction, let $G \Rightarrow_{p^*} M$ in $\mathrm{DPO}^{a/i}$ be an s-concurrent derivation through a proper s-concurrent rule p^*. Then p^* is constructed from an s-derivation $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ in $\mathrm{DPO}^{a/i}$. Now $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is an instance of the s-derivation in $\mathrm{DPO}^{a/i}$, that is, it is a pair of strongly sequentially s-dependent derivations in $\mathrm{DPO}^{a/i}$. An analogous argument holds for $\mathrm{DPO}^{i/i}$. □

Note that Theorem 8.13 does not hold either in $\mathrm{DPO}^{i/a}$ or $\mathrm{DPO}^{a/a}$. Since there are no restrictions for s, it is not the case that for every s-derivation there exists an s-derivation in $\mathrm{DPO}^{a/i}$. Thus, it is not the case that for every s-concurrent rule there is also a proper s-concurrent rule. Hence the first statement does not hold. Inspecting the proof of the second statement, one may see that the sequentially independent direct derivations need not be strongly sequentially independent; thus the second statement does not hold either.

9. Parallelism vs. concurrency

In this section we relate the results on parallelism and concurrency and show that the parallelism theorems in Section 7 are special cases of the concurrency theorems in Section 8. To this end, we define the notions of independence relation, sequential s-independence, s-parallel rule and derivation as special cases of the notions of dependence relation, sequential s-dependence, s-concurrent rule and derivation, respectively. The relations between the notions are summarized at the end of this section.

Definition 9.1. (Independence relation) A pair of graph morphisms $s = \langle K_1 \leftarrow S \rightarrow K_2 \rangle$ is an *independence relation* for $\langle p_1, p_2 \rangle$. It is *proper* if the composed morphisms $S \rightarrow K_1 \rightarrow R_1$ and $S \rightarrow K_2 \rightarrow R_2$ are injective.

In particular, $\epsilon = \langle K_1 \leftarrow \varnothing \rightarrow K_2 \rangle$ is an independence relation for $\langle p_1, p_2 \rangle$, called the *empty* independence relation for $\langle p_1, p_2 \rangle$.

By the following lemma, every independence relation $s = \langle K_1 \leftarrow S \rightarrow K_2 \rangle$ induces a dependence relation $\langle R_1 \leftarrow S \rightarrow L_2 \rangle$, where $S \rightarrow R_1$ and $S \rightarrow L_2$ denote the composed

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

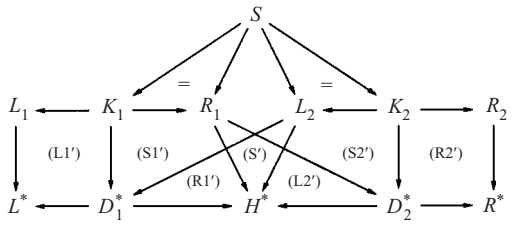


Fig. 30. The derivation $\Delta(p_1, s, p_2)$

morphisms $S \to K_1 \to R_1$ and $S \to K_2 \to L_2$, respectively. Somewhat ambiguously, the induced dependence relation is also denoted by s.

Lemma 9.2. (Independence) For every (proper) independence relation s for $\langle p_1, p_2 \rangle$, there are two (strongly) sequentially independent direct derivations $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$ such that the diagrams (S′), (S1′) and (S2′) in Figure 30 are pushouts. The derivation is denoted by $\Delta(p_1, s, p_2)$.

Proof. Let $s = \langle K_1 \leftarrow S \rightarrow K_2 \rangle$ be an independence relation for $\langle p_1, p_2 \rangle$ and let $S \to R_1$ and $S \to L_2$ denote the composed morphisms $S \to K_1 \to R_1$ and $S \to K_2 \to L_2$, respectively. Let (S′), (S1′) and (S2′) be pushouts of $R_1 \leftarrow S \to L_2$, $K_1 \leftarrow S \to L_2$ and $R_1 \leftarrow S \to K_2$, respectively. By the universal property of (S1′) and (S2′), there are unique morphisms $D_1^* \to H^*$ and $D_2^* \to H^*$ such that $L_2 \to D_1^* \to H^* = L_2 \to H^*$ and $R_1 \to D_2^* \to H^* = R_1 \to H^*$ and (R1′) and (L2′) commute. By the Decomposition Lemma A.19, the diagrams (R1′) and (L2′) are pushouts. Next, let (L1′) and (R2′) be pushouts of $L_1 \leftarrow K_1 \to D_1^*$ and $R_2 \leftarrow K_2 \to D_2^*$, respectively. Then the pushouts constitute two direct derivations $L^* \Rightarrow_{p_1} H^* \Rightarrow_{p_2} R^*$. By construction, these direct derivations are sequentially independent. If s is proper, then, by the injectivity of $S \to K_2 \to R_2$ and the pushout property of (S2′)+(R2′), we have $R_1 \to D_2^* \to R^*$ is injective, that is, the direct derivations are strongly sequentially independent. □

In order to relate sequential independence and sequential s-dependence, we consider the following subclass of sequential s-dependent direct derivations.

Definition 9.3. (Sequential s-independence) Let s be an independence relation for $\langle p_1, p_2 \rangle$. Two direct derivations $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ are *sequentially s-independent*, if diagram (S) in Figure 21 commutes and $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of $\Delta(p_1, s, p_2)$, where $H^* \to H$ is the unique morphism that exists by the universal property of the pushout (S′) in Figure 22. They are *strongly sequentially s-independent*, if diagram (S) commutes and $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is an instance of $\Delta(p_1, s, p_2)$.

Sequential independence and sequential s-independence for some independence relation s are related in a natural way.

Lemma 9.4. (Sequential independence and sequential s-independence) Two direct derivations are sequentially independent if and only if they are sequentially s-independent for some independence relation s. In $\mathrm{DPO}^{\mathrm{a/i}}$ and $\mathrm{DPO}^{\mathrm{i/i}}$, two direct derivations are strongly

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

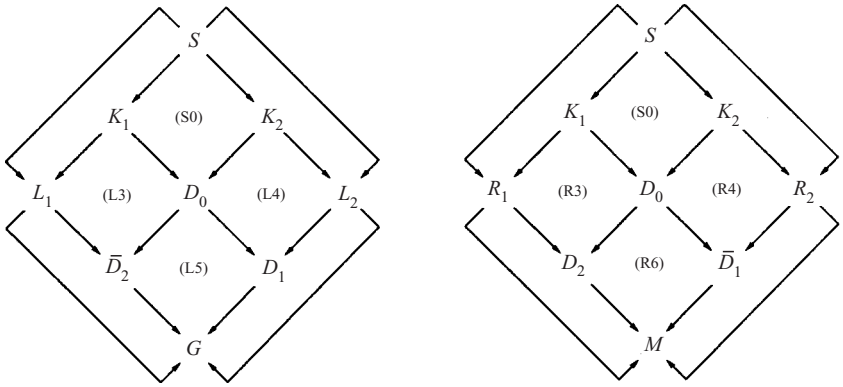


Fig. 31. Completed butterflies

sequentially independent if and only if they are strongly sequentially s-independent for some proper independence relation s.

Proof. 'Only if': Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be sequentially independent. Then the derivation can be decomposed as in the proof of Theorem 6.4 and all diagrams in Figure 10 are pushouts. The pushouts can be composed as butterflies (see Figure 16). Now construct (S0) as a pullback of $K_1 \to D_0 \leftarrow K_2$. Then we obtain the completed butterflies as shown in Figure 31.

The pair of morphisms $s = \langle K_1 \leftarrow S \to K_2 \rangle$ forms an independence relation for $\langle p_1, p_2 \rangle$. It is easy to show that $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of $\Delta(p_1, s, p_2)$, that is, the direct derivations are sequentially s-independent (see Figure 32).

This may be seen as follows. Since (S′) is a pushout and (S) commutes, there exists a unique morphism $H^* \to H$ such that $R_1 \to H^* \to H = R_1 \to H$ and $L_2 \to H^* \to H = L_2 \to H$. Since (S1′) is a pushout and (S0)+(L2′) commutes, there exists a unique morphism $D_1^* \to D_1$ such that $K_1 \to D_1^* \to D_1 = K_1 \to D_1$ and $L_2 \to D_1^* \to D_1 = L_2 \to D_1$ commutes. Since $K_1 \to D_1^* \to D_1 \to H = K_1 \to D_1^* \to H^* \to H$ and $L_2 \to D_1^* \to D_1 \to H = L_2 \to D_1^* \to H^* \to H$, and $\langle K_1 \to D_1^*, L_2 \to D_1^* \rangle$ is jointly surjective, $D_1^* \to D_1 \to H = D_1^* \to H^* \to H$. Accordingly, there exists a unique morphism $D_2^* \to D_2$ such that $K_2 \to D_2^* \to D_2 = K_2 \to D_2$ and $D_2^* \to D_2 \to H = D_2^* \to H^* \to H$. Since (L1′) is a pushout, (L1) commutes, and $K_1 \to D_1 = K_1 \to D_1^* \to D_1$, there exists a unique morphism $L^* \to G$ such that $L_1 \to L^* \to G = L_1 \to G$ and $D_1^* \to L^* \to G = D_1^* \to D_1 \to G$. Accordingly, there exists a unique morphism $R^* \to M$ such that $R_2 \to R^* \to M = R_2 \to M$ and $D_2^* \to R^* \to M = D_2^* \to D_2 \to M$. By the Decomposition Lemma A.19, the diagrams (L1″), (R1″), (L2″), and (R2″) are pushouts, that is, $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of $\Delta(p_1, s, p_2)$ and hence sequentially s-independent.

If $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is in $\mathrm{DPO}^{\mathrm{a/i}}$, then $\Delta(p_1, s, p_2)$ is in $\mathrm{DPO}^{\mathrm{a/i}}$ and $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is an instance of it: by the injectivity of $L_1 \to G$ and $L_2 \to H$, $L_2 \to D_1 \to G$ is injective. By the injectivity of $K_1 \to L_1$ and $K_2 \to L_2$, all diagrams in the left of Figure 31 are pullbacks. Consequently, the composite diagram ((S0)+(L3))+((L4)+(L5)) is a pullback. Moreover, the composed diagram (S1′)+(L1′) in Figure 30 is a pushout. By the Inheritance

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

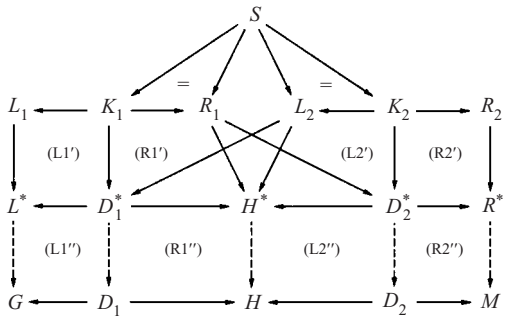


Fig. 32. Generalized instance of $\Delta(p_1, s, p_2)$

of injectivity A.17, the injectivity of $L_1 \to G$ and $L_2 \to D_1 \to G$ implies the injectivity of $L^* \to G$. Furthermore, the injectivity of $L^* \to G$ implies the injectivity of the other vertical morphisms, that is, $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is an instance of $\Delta(p_1, s, p_2)$.

Furthermore, by the injectivity of the morphisms $K_i \to L_i$ and $L_i \to G$ for $i = 1, 2$, all morphisms in the left of Figure 31 are injective. In particular, $S \to K_1$ and $S \to K_2$ are injective. Since $S \to K_2$ and $K_2 \to D_0 \to D_2$ in the right of Figure 31 are injective and (S0)+(R3) commutes, $S \to K_1 \to R_1$ is injective. If now $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is strongly sequentially independent, that is, $R_1 \to D_2 \to M$ is injective, then $S \to K_2 \to R_2$ is injective because the composite diagram ((S0)+(R3))+((R4)+(R6)) in the right of Figure 31 commutes. Hence, s is a proper independence relation.

'If': Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ be sequentially s-independent for some independence relation s. Then it is a generalized instance of $\Delta(p_1, s, p_2)$ (see Figure 32). Now define $R_1 \to D_2 = R_1 \to D_2^* \to D_2$ and $L_2 \to D_1 = L_2 \to D_1^* \to D_1$. Then $R_1 \to D_2 \to H = R_1 \to H$ and $L_2 \to D_1 \to H = L_2 \to H$, that is, the direct derivations are sequentially independent.

If $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{a/i}$ is strongly sequentially s-independent for some proper independence relation s, then $S \to K_1 \to R_1$ and $S \to K_2 \to L_2$ are injective, $\Delta(p_1, s, p_2)$ is in $\mathrm{DPO}^{a/i}$, and $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is an instance of it. Since (S2′)+(R2′) in Figure 30 is a pushout and $S \to K_2 \to R_2$ is injective, $R_1 \to D_2^* \to R^*$ is injective. Since $R^* \to M$ is injective and $R_1 \to D_2 \to M = R_1 \to D_2^* \to R^* \to M$, the morphism $R_1 \to D_2 \to M$ is injective, that is, the direct derivations are strongly sequentially independent. □

In order to relate parallel rules and s-concurrent rules, we consider the following subclass of s-concurrent rules.

Definition 9.5. (s-parallel rules and derivations) Let s be an independence relation for $\langle p_1, p_2 \rangle$. The *s-parallel rule* of $\langle p_1, p_2 \rangle$, denoted by $p_1 +_s p_2$, is the s-concurrent rule constructed from $\Delta(p_1, s, p_2)$. A direct derivation through an s-parallel rule is called an *s-parallel direct derivation.*

Example 9.6. Let S be the graph with a single node, and p_1 and p_2 be the rules in Figure 13. Then for every pair of morphisms, $s = \langle K_1 \leftarrow S \to K_2 \rangle$ is a proper independence relation for $\langle p_1, p_2 \rangle$, $\Delta(p_1, s, p_2)$ is the s-derivation $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in Figure 13, and the

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

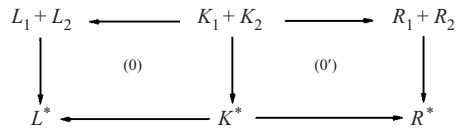


Fig. 33. A quotient rule of $p_1 + p_2$

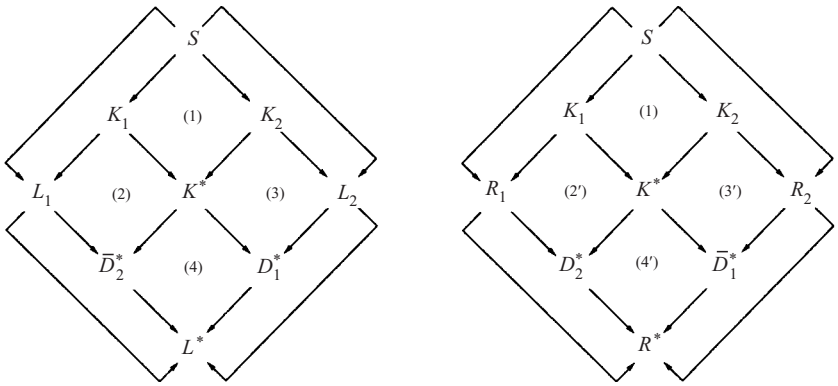


Fig. 34. Completed butterflies I

s-parallel rule constructed from $\Delta(p_1, s, p_2)$ is the proper quotient rule of $p_1 + p_2$ presented in Example 7.2.

Lemma 9.7. (Quotients of parallel rules vs. s-parallel rules) A rule is a proper quotient rule of $p_1 + p_2$ if and only if it is the s-parallel rule of $\langle p_1, p_2 \rangle$ for some proper independence relation s. Moreover, the parallel rule $p_1 + p_2$ is an ϵ-parallel rule of $\langle p_1, p_2 \rangle$ and the s-parallel rule of $\langle p_1, p_2 \rangle$ (for some independence relation s) is a quotient rule of $p_1 + p_2$.

Proof. 'Only if': Let p^* be a proper quotient rule of $p_1 + p_2$. Then the diagrams (0) and (0′) in Figure 33 are pushouts. By the Butterfly Lemma A.26, there exist decompositions of (0) and (0′) into pushouts (2), (3), (4) and (2′), (3′), (4′), respectively (see Figure 34).

Now let (1) be a pullback of $K_1 \to K^* \leftarrow K_2$. Then $s = \langle K_1 \leftarrow S \to K_2 \rangle$ is an independence relation for $\langle p_1, p_2 \rangle$. Moreover, p^* is the s-parallel rule of $\langle p_1, p_2 \rangle$. This may be seen as follows. Since p^* is proper, the morphism $L_i \to L_1 + L_2 \to L^*$ for $i = 1, 2$ is injective. Therefore, the morphism $K_i \to K^* = K_i \to K_1 + K_2 \to K^*$ for $i = 1, 2$ is injective. Since $K_1 + K_2 \to K^*$ is surjective, the pair $\langle K_1 \to K^*, K_2 \to K^* \rangle$ is jointly surjective. By the Pullback–Pushout Lemma A.12, diagram (1) is a pushout as well. Now all diagrams in Figure 34 are pushouts, and, by the Composition Lemma A.18, all composed diagrams are pushouts.

Now construct (4″) as a pushout of $D_1^* \leftarrow K^* \to D_2^*$. Then all diagrams in Figure 35 are pushouts and, by the Composition Lemma A.18, all composed diagrams are pushouts. By the Injectivity Lemma A.14 and the Pushout–Pullback Lemma A.11, the injectivity of $K_2 \to L_2$ implies the injectivity of $K^* \to D_2^*$ and (4″) is a pullback, as well.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

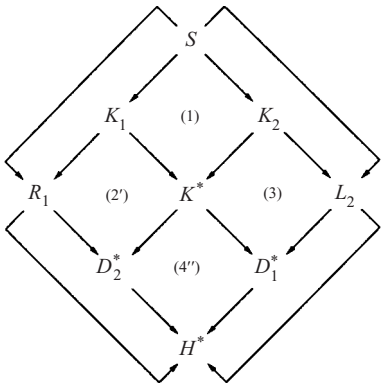


Fig. 35. Completed butterflies II

Composing the diagrams in a new fashion, we obtain the derivation $\Delta(p_1, s, p_2)$ as in Figure 30, and p^* is the s-parallel rule constructed from it. Moreover, the independence relation s is proper, because p^* is a proper quotient of $p_1 + p_2 \colon S \to K^* \to R^*$ is injective, and, by the commutativity of the diagram on the right of Figure 34, $S \to K_1 \to R_1$ and $S \to K_2 \to R_2$ are injective.

If p^* is the parallel rule $p_1 + p_2$, then $K^* = K_1 + K_2$, $S = \varnothing$ and p^* is the ϵ-parallel rule of $\langle p_1, p_2 \rangle$.

'If': Let p^* be an s-parallel rule for $\langle p_1, p_2 \rangle$ for some independence relation s. Then p^* is constructed from $\Delta(p_1, s, p_2)$. The pushout (S′) in Figure 30 may be decomposed into commutative diagrams (1), (2′), (3) and (4″) as in Figure 35 where (4″) is a pullback and (1)+(2′) and (1)+(3) are pushouts. By the Decomposition Lemma A.19, (2′)+(4″) and (3)+(4″) are pushouts. Moreover, $K^* \to D_1^*$ is injective. By the Triple-Pushout Lemma A.21, (2′), (3′) and (4″) are pushouts. Finally, by the Special Decomposition Lemma A.20, the pushout property of (1)+(3) and (3) implies the pushout property of (1). Thus all diagrams in Figure 35 are pushouts.

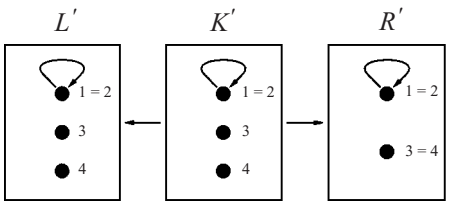
Now the diagrams may be rearranged and the pushouts (L1′) and (R2′) in Figure 30 may be decomposed into pushouts (2), (4), (3′) and (4′). Then all diagrams in Figure 34 are pushouts and, by the Composition Lemma A.18, all composed diagrams are pushouts. By the Butterfly Lemma A.26, the diagrams (0) and (0′) in Figure 34 are pushouts. Since, by the pushout property of (1), the pair $\langle K_1 \to K^*, K_2 \to K^* \rangle$ is jointly surjective, the morphism $K_1 + K_2 \to K^*$ is surjective. As a consequence, p^* is a quotient rule of $p_1 + p_2$.

If s is proper, then the injectivity of $S \to K_1 \to R_1$ and $S \to K_2 \to R_2$ implies the injectivity of $S \to K_1$ and $S \to K_2$. As a consequence, the morphisms $S \to K_1 \to L_1$ and $S \to K_2 \to L_2$ in the left of Figure 34 are injective. By the pushout property of the diagram, $L_1 \to L^* = L_1 \to L_1 + L_2 \to L^*$ and $L_2 \to L^* = L_2 \to L_1 + L_2 \to L^*$ are injective. Furthermore, the injectivity of $S \to K_1 \to R_1$ and $S \to K_2 \to R_2$ implies the injectivity of $R_1 \to R^*$ and $R_2 \to R^*$ in the right of Figure 34. Therefore, $S \to K^* \to R^*$ is injective. Thus p^* is a proper quotient rule of $p_1 + p_2$. □

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

The following counter-example demonstrates that not every quotient rule of $p_1 + p_2$ is the s-parallel rule of $\langle p_1, p_2 \rangle$ for some independence relation s.

Example 9.8. Consider the rules p_1 and p_2 in Figure 12. Then the rule p' below is a non-proper quotient rule of $p_1 + p_2$. Note that there is no independence relation s such that p' is the s-parallel rule of $\langle p_1, p_2 \rangle$: the empty relation yields the parallel rule and every relation s that relates at least one node of K_1 with a node of K_2 yields an s-parallel rule in which these nodes are identified.



Lemma 9.7 provides an alternative method for constructing the set of all proper quotient rules of $p_1 + p_2$. For every graph S with morphisms $S \to K_i$ such that $S \to K_i \to R_i$ is injective, we 'glue' the left-hand sides, the right-hand sides, and the interfaces of p_1 and p_2 according to S. We construct the left-hand side, the right-hand side, and the interface of the new rule as the pushout objects of $L_1 \leftarrow S \to L_2$, $R_1 \leftarrow S \to R_2$, and $K_1 \leftarrow S \to K_2$, respectively, where $S \to L_i = S \to K_i \to L_i$ and $S \to R_i = S \to K_i \to R_i$. Then there exist unique morphisms from the interface to the left- and the right-hand side. The rule obtained in this way is a proper quotient rule of $p_1 + p_2$ (see the proof of Lemma 9.7).

We conclude this section with the following nice relationship.

Corollary 9.9. (Parallelism vs. Concurrency) The Parallelism Theorem I (II) is a special case of the Concurrency Theorem I (II).

Proof. Let $x \in \{\mathrm{a}, \mathrm{i}\}$.

(1) Let $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{x/\mathrm{a}}$ ($\mathrm{DPO}^{x/\mathrm{i}}$) be (strongly) sequentially independent. By Lemma 9.4, the direct derivations are (strongly) sequentially s-independent for some (proper) independence relation s. Therefore, $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is a generalized instance of $\Delta(p_1, s, p_2)$. By the proof of Theorem 8.8 (8.13), there is an s-parallel derivation $G \Rightarrow_{p_1 +_s p_2} M$ in $\mathrm{DPO}^{x/\mathrm{a}}$ ($\mathrm{DPO}^{x/\mathrm{i}}$). By Lemma 9.7, the rule $p_1 +_s p_2$ is in $\mathrm{Q}(p_1 + p_2)$ ($\mathrm{PQ}(p_1 + p_2)$). By the Quotient Lemma, there is also a derivation $G \Rightarrow_{p_1 + p_2} M$ in $\mathrm{DPO}^{x/\mathrm{a}}$.

(2) Let $G \Rightarrow_{p_1 + p_2} M$ in $\mathrm{DPO}^{x/\mathrm{a}}$ ($G \Rightarrow_{\mathrm{PQ}(p_1 + p_2)} M$ in $\mathrm{DPO}^{x/\mathrm{i}}$). By Lemma 9.7, the used rule is an s-parallel rule for some (proper) independence relation s. Hence, the rule is constructed from the derivation $\Delta(p_1, s, p_2)$. Therefore, the derivation $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ in $\mathrm{DPO}^{x/\mathrm{a}}$ ($\mathrm{DPO}^{x/\mathrm{i}}$) constructed in the proof of Theorem 8.8 (8.13) is a generalized instance (respectively, an instance) of $\Delta(p_1, s, p_2)$. Hence $G \Rightarrow_{p_1} H \Rightarrow_{p_2} M$ is (strongly) sequentially s-independent, and, by Lemma 9.4, (strongly) sequentially independent. □

The relations between the notions introduced in Sections 7, 8 and 9 may be summarized

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

as follows:

Parallelism		Parallelism/Concur.		Concurrency
seq. independent	$\Leftrightarrow$	seq. s-independent for some independence relation s	$\Rightarrow$	seq. s-dependent for some dependence relation s
strongly seq. independent in $\mathrm{DPO}^{x/\mathrm{i}}$	$\Leftrightarrow$	strongly seq. s-independent for some proper independence relation s	$\Rightarrow$	strongly seq. s-dependent for some dependence relation s
parallel rule quotient of a parallel rule	$\Rightarrow$ $\Leftarrow$	s-parallel rule for some independence relation s	$\Rightarrow$	s-concurrent rule for some dependence relation s
proper quotient of a parallel rule	$\Leftrightarrow$	s-parallel rule for some proper independence relation s	$\Rightarrow$	s-concurrent rule for some dependence relation s

10. Conclusion

We have shown that injective matching makes double-pushout graph transformation more expressive with respect to both, the generative power of grammars without non-terminals and the computability of functions by convergent graph transformation systems. These findings also justify the practice in programming systems for graph transformation, like PROGRES (Schürr *et al.* 1999), to match rules injectively.

The classical independence, parallelism and concurrency 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 counter-example:

	$\mathrm{DPO}^{\mathrm{i/a}}$	$\mathrm{DPO}^{\mathrm{a/a}}$	$\mathrm{DPO}^{\mathrm{i/i}}$	$\mathrm{DPO}^{\mathrm{a/i}}$
char. of parallel independence	yes	yes	yes	yes
char. of sequential 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
parallelism I	yes	yes	no	no
parallelism I/II	yes	yes	yes	yes
concurrency I	yes	yes	no	no
concurrency I/II	yes	yes	yes	yes

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

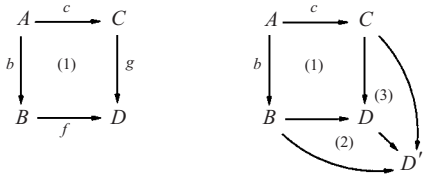


Fig. 36. A pushout diagram (on the left)

The commutativity results II & III are proved for $DPO^{a/i}$; the corresponding results for the other variants follow from the commutativity result I and the Simulation Theorem. The parallelism and concurrency results I hold for $DPO^{i/a}$ and $DPO^{a/a}$; the parallelism and concurrency results II hold for $DPO^{i/i}$ and $DPO^{a/i}$. This gives a complete picture of positive results.

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 so-called single-pushout approach (Löwe 1993). In Habel *et al.* (1996), 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 Habel *et al.* (1996) requires more information on the given direct derivations than our definition of strong sequential independence.

A further topic is to consider so-called high-level replacement systems, that is, double-pushout derivations in arbitrary categories. We just mention that our results do not follow from those in Ehrig *et al.* (1991), where non-injective rules and injective matching morphisms are not considered.

Finally, the computability model for graph functions introduced in Subsection 4.2 should be further investigated. Since this model does not cover all (in the ordinary sense) computable functions on graphs, there is also the quest to find a model based on the double-pushout approach that has universal computational power[†].

Acknowledgement

We are grateful to the anonymous referees for providing helpful comments.

Appendix A. Facts about pushouts and pullbacks

This appendix summarizes facts about pushouts and pullbacks that are used in this paper. Readers interested in the category-theoretic background of these concepts may consult, for example, Adamek *et al.* (1990).

Definition A.1. (Pushout) Given two graph morphisms $A \to B$ and $A \to C$, a graph D together with two graph morphisms $B \to D$ and $C \to D$ is a *pushout* of $A \to B$ and $A \to C$ if the following conditions are satisfied:

[†] Note added at proof stage: such a model is given in Habel and Plump (2001).

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

1 Commutativity: $A \to B \to D = A \to C \to D$.
2 Universal property: For all graphs D' and graph morphisms $B \to D'$ and $C \to D'$ such that $A \to B \to D' = A \to C \to D'$, there is a unique morphism $D \to D'$ such that $B \to D \to D' = B \to D'$ and $C \to D \to D' = C \to D'$ (see the right part of Figure 36).

In this case the diagram on the left of Figure 36 will also be called a pushout and will be denoted by $ABCD$ or $ACBD$.

Graph pushouts can be constructed set-theoretically as follows.

Lemma A.2. (Pushout construction (Ehrig 1979)) Let $b : A \to B$ and $c : A \to C$ be graph morphisms. Then a pushout as in Figure 36 can be constructed as follows:

— $D = (B + C)/\approx$, separately for nodes and edges, where $\approx$ is the equivalence relation generated by the relation $\sim$ defined by $b(a) \sim c(a)$ for all a in A; $s_D([e]) = \underline{\text{if}}\ e \in E_B\ \underline{\text{then}}\ [s_B(e)]\ \underline{\text{else}}\ [s_C(e)]$ where $[e]$ for $e \in E_B + E_C$ is the equivalence class of e with respect to $\approx$; t_D is defined analogously to s_D; $l_D([v]) = \underline{\text{if}}\ v \in V_B\ \underline{\text{then}}\ l_B(v)\ \underline{\text{else}}\ l_C(v)$ where $[v]$ for $v \in V_B + V_C$ is the equivalence class of v with respect to $\approx$; m_D is defined analogously to l_D.
— $f : B \to D$ and $g : C \to D$ are the graph morphisms sending each element to its equivalence class, that is, $f(x) = [x]$ and $g(x) = [x]$, separately for nodes and edges.

Lemma A.3. (Uniqueness of pushouts (Adamek *et al.* 1990)) Let diagram (1) in Figure 36 be a pushout. A graph D' together with graph morphisms $B \to D'$ and $C \to D'$ is a pushout of $A \to B$ and $A \to C$ if and only if there is an isomorphism $D \to D'$ such that the triangles (2) and (3) in the right part of Figure 36 commute.

Definition A.4. (Pushout complement) Given two graph morphisms $A \to B$ and $B \to D$, a graph C together with two graph morphisms $A \to C$ and $C \to D$ is a *pushout complement* of $A \to B$ and $B \to D$ if diagram (1) in Figure 36 is a pushout.

Lemma A.5. (Uniqueness of pushout complements (Rosen 1975)) Let diagram (1) in Figure 36 be a pushout. If C' together with graph morphisms $A \to C'$ and $C' \to D$ is a pushout complement of $A \to B$ and $B \to D$, and $A \to B$ is injective, then there is an isomorphism $C \to C'$ such that $A \to C \to C' = A \to C'$ and $C \to C' \to D = C \to D$.

Definition A.6. Let $b : A \to B$, $c : A \to C$, $f : B \to D$ and $g : C \to D$ be graph morphisms as in Figure 36.

1 The morphism f is *injective up to* b if for all items x and x' in B with $x \neq x'$ and $f(x) = f(x')$, we have x and x' are in $b(A)$.
2 The pair $\langle f, g \rangle$ is *jointly surjective* if $f(B) \cup g(C) = D$.
3 The commutative diagram (1) satisfies the *chain-condition* if for all items x in B and y in C with $f(b) = g(c)$ there are items $a_1, \ldots, a_{2n+1}$ in A such that $b(a_1) = x$, $c(a_{2n+1}) = y$, $c(a_{2i-1}) = c(a_{2i})$ and $b(a_{2i}) = b(a_{2i+1})$ for $i = 1, \ldots, n$. The chain-condition is called *reduced* in the case $n = 0$.
4 The pair $\langle b, c \rangle$ is *monomorphic* if for all items a and a' in A, $b(a) = b(a')$ and $c(a) = c(a')$ implies $a = a'$.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

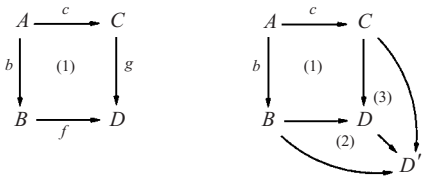


Fig. 37. A pullback

Lemma A.7. (Pushout characterization (Ehrig and Kreowski 1979)) Let diagram (1) in Figure 36 be commutative. Then (1) is a pushout if and only if:

1 f is injective up to b;
2 g is injective up to c;
3 $\langle f, g\rangle$ is jointly surjective; and
4 it satisfies the chain-condition.

For some proofs in Sections 6 and 8 we also need the dual concept to a pushout, which is introduced next.

Definition A.8. (Pullback) Given two graph morphisms $B \to A$ and $C \to A$, a graph D together with two graph morphisms $D \to B$ and $D \to C$ is a *pullback* of $B \to A$ and $C \to A$ if diagram (1) in Figure 37 commutes and if for all graphs D' and graph morphisms $D' \to B$ and $D' \to C$ such that $D' \to B \to A = D' \to C \to A$ there is a unique morphism $D' \to D$ such that the triangles (2) and (3) in Figure 37 commute.

Lemma A.9. (Pullback construction (Ehrig 1979)) Let $b : B \to A$ and $c : C \to A$ be graph morphisms. Then a pullback as in Figure 37 can be constructed as follows:

— $D = \{\langle x, y\rangle \in B \times C \mid b(x) = c(y)\}$, separately for nodes and edges; $s_D(\langle x, y\rangle) = \langle s_B(x), s_C(y)\rangle$ for $\langle x, y\rangle \in E_B \times E_C$; t_D is defined analogously to s_D; $l_D(\langle x, y\rangle) = l_B(x)$ for $\langle x, y\rangle \in V_B \times V_C$; m_D is defined analogously to l_D.
— $f : D \to B$ and $g : D \to C$ are the domain restrictions of the projections from $B \times C$, that is, $f(\langle x, y\rangle) = x$ and $g(\langle x, y\rangle) = y$, separately for nodes and edges.

Lemma A.10. (Pullback characterization (Ehrig and Kreowski 1979)) Let diagram (1) in Figure 37 be commutative. Then (1) is a pullback if and only if it satisfies the reduced chain-condition and $\langle f, g\rangle$ is monomorphic.

The following two lemmata are immediate consequences of the pullback and pushout characterization, respectively.

Lemma A.11. (Pushout–Pullback Lemma (Ehrig and Kreowski 1979)) Let diagram (1) in Figure 36 be a pushout. Then (1) is a pullback if one of the following conditions is satisfied:

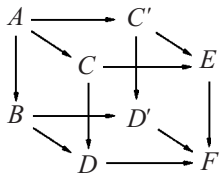
1 b or c is injective.
2 f or g is injective and $\langle b, c\rangle$ is monomorphic.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Lemma A.12. (Pullback–Pushout Lemma (Ehrig and Kreowski 1979)) Let diagram (1) in Figure 36 be a pullback. Then (1) is a pushout if one of the following conditions is satisfied:

1 f and g are injective and $\langle f, g\rangle$ is jointly surjective.
2 $\langle f, g\rangle$ is a pushout pair, that is, conditions 1–3 of the Pushout Characterization A.7 are satisfied.

Lemma A.13. (Cube Pullback–Pushout Lemma) Consider the following commutative diagram:



If the top and the bottom squares are pullbacks, the front and right squares are pushouts, $\langle C \to D, C \to E\rangle$ is monomorphic, and $C' \to E$ is injective, then the left and the back squares are pushouts.

A restricted version of the Cube Pullback–Pushout Lemma A.13 in which all morphisms in the top and bottom diagram are required to be injective is presented in Ehrig *et al.* (1991).

Proof. We show that the left square is a pushout. Then, by the Special Decomposition Lemma A.20, the back square is a pushout as well.

Since the right square is a pushout and $C' \to E$ is injective, the right square is a pullback, as well (Pushout–Pullback Lemma A.11). Since the top, the right and the bottom squares are pullbacks, the left square is a pullback (Decomposition Lemma A.19). Now, by the Pullback–Pushout Lemma A.12, the left square is a pushout. This may be seen as follows:

1 $B \to D$ is injective: the injectivity of $C' \to E$ implies the injectivity of $D' \to F$ and $B \to D$ (Injectivity Lemma A.14).
2 $C \to D$ is injective up to $A \to C$ (see Definition A.6): let x and y be distinct items in C that are identified by $C \to D$, and x' and y' be the images of x and y in E. Since, by assumption, the pair $\langle C \to D, C \to E\rangle$ is monomorphic, x' and y' are distinct. Since the front square is a pushout, x' and y' are identified by $E \to F$. Since the right square is a pushout, there are preimages of x' and y' in C'. Since the top square is a pullback, there are preimages of x and y in A.
3 $\langle B \to D, C \to D\rangle$ is jointly surjective (see Definition A.6): let x be an item in D and x' be the image of x in F. Since the right square is a pushout, $\langle D' \to F, E \to F\rangle$ is jointly surjective. If x' is in the image of D', then, by the pullback property of the bottom square, x is in the image of B. If x' is in the image of E, by the pushout property of the front square, more precisely, by the chain condition (see Definition A.6), x is in the image of C. □

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Lemma A.14. (Injectivity) Consider the diagram in Figure 36.

1 If (1) is a pushout and $A \to B$ is injective, $C \to D$ is injective.
2 If (1) is a pullback and $C \to D$ is injective, $A \to B$ is injective.

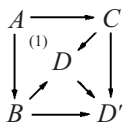
Proof. The first property follows from the corresponding result for sets; the second property is valid for any category (see, for example, Adamek *et al.* (1990, Proposition 11.18)). □

Lemma A.15. (Surjectivity) Consider the diagram in Figure 36.

1 If (1) is a pushout and $A \to B$ is surjective, $C \to D$ is surjective.
2 If (1) is a pullback and $C \to D$ is surjective, $A \to B$ is surjective.

Proof. The first property is valid for any category (see, for example, Adamek *et al.* (1990, Proposition 11.18)); the second property follows from the corresponding result for sets. □

Lemma A.16. (Inheritance of the pullback property (Adamek *et al.* 1990)) Suppose that the following diagram commutes:



1 If the outer square is a pullback, so is (1).
2 If (1) is a pullback and $D \to D'$ is injective, the outer square is a pullback.

Lemma A.17. (Inheritance of injectivity) Consider the diagram above. If the outer square is a pullback, (1) is a pushout, $B \to D'$ and $C \to D'$ are injective, then $D \to D'$ is injective.

Proof. Suppose, to the contrary, that there are two distinct items x and y in D that are identified by $D \to D'$. By the pushout property of (1), $\langle B \to D, C \to D \rangle$ is jointly surjective (Pushout Characterization A.7). Now three cases have to be considered.

Case 1. x and y are both in the image of B. Then the preimages of x and y must be identified by $B \to D'$, which contradicts the injectivity of $D \to D'$.

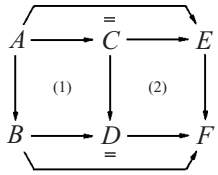
Case 2. x and y are both in the image of C. This is analogous to Case 1.

Case 3. x and y are in the image of B and C, respectively. Then there are preimages x' of x in B and y' of y in C with $(B \to D')(x') = (C \to D')(y')$. By the pullback property of the outer square, there exists a common preimage a in A that is mapped to x' and y', respectively (Pullback construction A.9). By the pushout property of (1), the images of x' and y' in D are equal, which is a contradiction to the distinctness of x and y.

This completes the proof. □

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

In the following, we consider the following commutative diagram:



Lemma A.18. (Composition (Adamek *et al.* 1990)) If diagrams (1) and (2) are pushouts (pullbacks), then the composite diagram (1)+(2) is a pushout (pullback).

Lemma A.19. (Decomposition (Adamek *et al.* 1990))

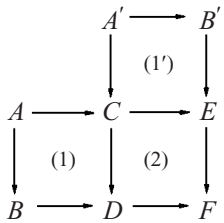
1 If (1)+(2) and (1) are pushouts, then (2) is a pushout.
2 If (1)+(2) and (2) are pullbacks, then (1) is a pullback.

Lemma A.20. (Special decomposition (Ehrig and Kreowski 1979))

1 If (1)+(2) and (2) are pushouts and $C \to E$ is injective, then (1) is a pushout.
2 If (1)+(2) is a pullback, (1) is a pushout, and $E \to F$ is injective, then (2) is a pullback.

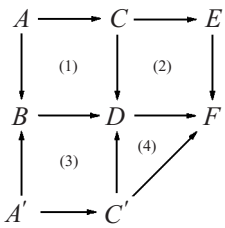
Beside the Special Decomposition Lemma A.20, we present a so-called Triple-Pushout and a Twisted-Triple-Pushout Lemma in which the factorization of pushouts is triggered by a pullback (and a pushout) construction.

Lemma A.21. (Triple-Pushout Lemma (Ehrig and Kreowski 1979)) Suppose that the following diagram commutes.



If (1)+(2) and (1′)+(2) are pushouts, (2) is a pullback, and $C \to D$ or $C \to E$ are injective, then (1), (1′), and (2) are pushouts.

Lemma A.22. (Twisted-Triple-Pushout Lemma) Suppose that the following diagram commutes.



If (1)+(2) is a pushout, (2) is a pullback, (3) is a pushout, (3)+(4) is a pullback and $A \to B$ or $D \to F$ is injective, then (1) and (2) are pushouts.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

A restricted version of the Twisted-Triple Pushout Lemma A.22 in which $A \to B$ is required to be injective is proved in Ehrig *et al.* (1991).

Proof. We will show that (1) is a pushout by the Pushout characterization A.7. Then, by the Decomposition Lemma A.19, (2) is a pushout as well.

1 $B \to D$ is injective up to $A \to B$. By the pushout property of (1)+(2), we have $B \to D \to F$ and $B \to D$ are injective up to $A \to B$.
2 $C \to D$ is injective up to $A \to C$. If $A \to B$ is injective, then the pushout property of (1)+(2) and the pullback property of (2) imply the injectivity of $E \to F$ and $C \to D$ (Injectivity Lemma A.14). If $D \to F$ is injective, the pullback property of (2) implies the injectivity of $C \to E$ (Injectivity Lemma A.14). Now let x and y be two distinct items in C that are identified by $C \to D$, and x' and y' be the images of x and y in E. By the pullback property of (2), we have $\langle C \to D, C \to E\rangle$ is monomorphic and (2) is commutative. Hence, x' and y' are distinct items that are identified by $E \to F$. By the pushout property of (1)+(2), we have $E \to F$ is injective up to $A \to C \to E$, that is, there are preimages of x' and y' in A. Since $C \to E$ is injective, the preimages of x' and y' are preimages of x and y, too.
3 $\langle B \to D, C \to D\rangle$ are jointly surjective. By the pushout property of (3), $\langle B \to D, C' \to D\rangle$ is jointly surjective, that is, every item d in D has a preimage in B or a preimage c' in C'. In the first case, we are ready. In the second case, let f be the image of d in F. By the pushout property of (1)+(2), f has a preimage b in B or a preimage e in E.
Case 1. f has a preimage b in B. By the pullback property of (3)+(4), there is a common preimage of b and c' in A' and, by the commutativity of (3), b is mapped to d. Hence, b is a preimage of d in B.
Case 2. f has a preimage e in E. By the pullback property of (2), there is a common preimage of d and e in C.
4 Diagram (1) satisfies the chain-condition. If $A \to B$ is injective, then, by the Pushout–Pullback Lemma A.11 and the Decomposition Lemma A.19, (1)+(2) and (1) are pullbacks, and, by the Pullback characterization A.10, (1) satisfies the reduced chain-condition. If $D \to F$ is injective, $C \to E$ is injective. Now let b and c be items in B and C with common image d in D, and let e and f be the images of c and d in E and F, respectively. By the pushout property of (1)+(2), we have (1)+(2) satisfies the chain-condition. Therefore, there exists a chain connecting b and e, that is, there are items $a_1, \ldots, a_{2n+1}$ in A such that $(A \to B)(a_1) = b$, $(A \to C \to E)(a_{2n+1}) = e$, $(A \to B)(a_{2i-1}) = (A \to B)(a_{2i})$ and $(A \to C \to E)(a_{2i}) = (A \to C \to E)(a_{2i+1})$ for $i = 1, \ldots, n$. Since $C \to E$ is injective, $(A \to C)(a_{2n+1}) = d$ and $(A \to C)(a_{2i}) = (A \to C)(a_{2i+1})$ for $i = 1, \ldots, n$, that is, there exists a chain connecting b and c. □

The proof of the Twisted-Triple-Pushout Lemma A.22 shows that it is not necessary to assume that (3) is a pushout; we only need that the pair $\langle B \to D, C' \to D\rangle$ is jointly surjective.

In the following, we will consider epi-mono factorizations of pairs of morphisms. This allows us to switch from arbitrary pushouts to pushouts with monomorphic pairs.

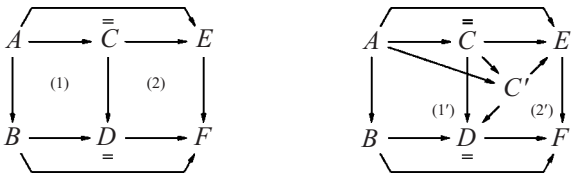
https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Lemma A.23. (Epi-mono factorization) There exists an epi-mono factorization, for every pair of morphisms $\langle b : A \to B, c : A \to C \rangle$, that is, there is a surjective morphism $e : A \to A'$ and a monomorphic pair $\langle b' : A' \to B, c' : A' \to C \rangle$ such that $b' \circ e = b$ and $c' \circ e = c$.

Proof. Let $A' = A/\approx$ separately for nodes and edges, where $\approx$ is the equivalence relation defined by $a \approx a'$ if and only if $b(a) = b(a')$ and $c(a) = c(a')$ for all a and a' in A; $s_{A'}([e]) = [s_A(e)]$ where $[e]$ for $e \in E_A$ is the equivalence class of e with respect to $\approx$; $t_{A'}$ is defined analogously to $s_{A'}$; $l_{A'}([v]) = l_A(v)$ where $[v]$ for $v \in V_A$ is the equivalence class of v with respect to $\approx$; $m_{A'}$ is defined analogously to $l_{A'}$. Let $e : A \to A'$ be the graph morphism sending each element to its equivalence class, that is, $e(x) = [x]$, and $b' : A' \to B$ and $c' : A' \to C$ be the graph morphisms with $b'([x]) = b(x)$ and $c'([x]) = c(x)$, separately for nodes and edges. Then e is surjective, b' and c' are well-defined, $\langle b', c' \rangle$ is monomorphic, $b' \circ e = b$ and $c' \circ e = c$. □

Analogously, one can show that for every morphism $b : A \to B$, there is a surjective morphism $e : A \to A'$ and an injective morphism $b' : A' \to B$ such that $b' \circ e = b$.

Lemma A.24. (Monomorphic pairs) Consider the diagrams below. If (1) and (2) are pushouts, then there exists a factorization of the composite diagram (1)+(2) into pushouts (1′) and (2′) such that the pair $\langle C' \to D, C' \to E \rangle$ is monomorphic.



Proof. Let (1) and (2) be pushouts. Then there is an epi-mono factorization of the pair $\langle C \to D, C \to E \rangle$, that is, a surjective graph morphism $C \to C'$ and a monomorphic pair $\langle C' \to D, C' \to E \rangle$ such that $C \to C' \to D = C \to D$ and $C \to C' \to E = C \to E$. Since diagram (2) commutes and $C \to C'$ is surjective, (2′) commutes. By the pushout property of (2) and the dual of Lemma A.16, (2′) is a pushout.

Now let $A \to C' = A \to C \to C'$. Then diagram (1′) commutes and, by the pushout property of (1) and the surjectivity of $C \to C'$, (1′) is a pushout. This may be seen as follows. Let D' be an arbitrary graph and $B \to D'$ and $C' \to D'$ be graph morphisms such that $A \to B \to D' = A \to C' \to D'$. Define $C \to D' = C \to C' \to D'$. Then $A \to B \to D' = A \to C \to D'$, and, by the pushout property of (1), there is a unique morphism $D \to D'$ such that $B \to D \to D' = B \to D'$ and $C \to D \to D' = C \to D'$. Now $C \to C' \to D \to D' = C \to D \to D' = C \to D' = C \to C' \to D'$, and, by the surjectivity of $C \to C'$, $C' \to D \to D' = C' \to D'$. □

Note that there is a significant difference between the Factorization Lemma 2.1 in Ehrig and Kreowski (1979) and the Monomorphic-Pair Lemma: while Ehrig and Kreowski start with pushout (1)+(2) and look for conditions under which (1) and (2) are pushouts separately, we start with pushouts (1) and (2) and look for a factorization of (1)+(2) satisfying suitable properties.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

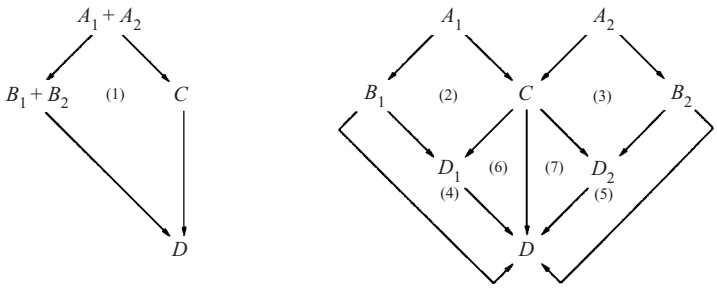


Fig. 38. Illustration for the Butterfly Lemma

Lemma A.25. (Lifting of injectivity) Consider the diagrams in Lemma A.24. Let (1) be a pullback, (2) be a pushout and $B \to D$ be injective. Then the injectivity of $A \to C \to E$ implies the injectivity of $B \to D \to F$.

Proof. Suppose, to the contrary, that there are two distinct items x and y in B that are identified by $B \to D \to F$. Since $B \to D$ is injective, the images of x and y in D are distinct and are identified by $D \to F$. Since (2) is a pushout, there are preimages x' and y' in C that are identified by $C \to E$ (Pushout Characterization A.7). Since (1) is a pullback, there are preimages of x' and y' in A. Therefore the morphism $A \to C \to E$ is not injective, which is a contradiction. □

Lemma A.26. (Butterfly Lemma (Kreowski 1977)) Diagram (1) on the left of Figure 38 is a pushout if and only if there exists a decomposition into diagrams (2) to (7) on the right of Figure 38 such that the diagrams (2), (3) and (6)+(7) are pushouts.

References

Adámek, J., Herrlich, H. and Strecker, G. (1990) *Abstract and Concrete Categories*, John Wiley.

Baader, F. and Nipkow, T. (1998) *Term Rewriting and All That*, Cambridge University Press.

Corradini, A., Montanari, U., Rossi, F., Ehrig, H., Heckel, R. and Löwe, M. (1997) Algebraic approaches to graph transformation. Part I: Basic concepts and double pushout approach. In: Rozenberg, G. (ed.) *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, World Scientific, chapter 3, 163–245.

Ehrig, H. (1979) Introduction to the algebraic theory of graph grammars. In: Proc. Graph-Grammars and Their Application to Computer Science and Biology. *Springer-Verlag Lecture Notes in Computer Science* **73** 1–69.

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

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

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

Ehrig, H., Habel, A. and Rosen, B. K. (1986) Concurrent transformations of relational structures. *Fundamenta Informaticae* **IX** 13–50.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press

Ehrig, H. and Kreowski, H.-J. (1976) Parallelism of manipulations in multidimensional information structures. In: Proc. Mathematical Foundations of Computer Science. *Springer-Verlag Lecture Notes in Computer Science* **45** 284–293.

Ehrig, H. and Kreowski, H.-J. (1979) Pushout-properties: An analysis of gluing constructions for graphs. *Mathematische Nachrichten* **91** 135–149.

Ehrig, H., Pfender, M. and Schneider, H. J. (1973) Graph grammars: An algebraic approach. In: *Proc. IEEE Conference on Automata and Switching Theory* 167–180.

Ehrig, H. and Rosen, B. K. (1976) Commutativity of independent transformations on complex objects. IBM Research Report RC 6251, Yorktown Heights.

Habel, A., Heckel, R. and Taentzer, G. (1996) Graph grammars with negative application conditions. *Fundamenta Informaticae* **26** 287–313.

Habel, A., Müller, J. and Plump, D. (2000) Double-pushout approach with injective matching. In: Proc. Theory and Application of Graph Transformations. *Springer-Verlag Lecture Notes in Computer Science* **1764** 103–116.

Habel, A. and Plump, D. (2001) Computational Completeness of Programming Languages Based on Graph Transformation. In: Foundations of Software Science and Computation Structures (FOSSACS 2001). *Springer-Verlag Lecture Notes in Computer Science* **2030** 230–245.

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

Kreowski, H.-J. (1977) *Manipulationen von Graphmanipulationen*, Doctoral dissertation, Fachbereich Informatik, Technische Universität Berlin.

Löwe, M. (1993) Algebraic approach to single-pushout graph transformation. *Theoretical Computer Science* **109** 181–224.

Plump, D. (1993) Hypergraph rewriting: Critical pairs and undecidability of confluence. In: Sleep, R., Plasmeijer, R. and van Eekelen, M. (eds.) *Term Graph Rewriting: Theory and Practice*, John Wiley, chapter 15, 201–213.

Plump, D. (1995) On termination of graph rewriting. In: Proc. Graph-Theoretic Concepts in Computer Science. *Springer-Verlag Lecture Notes in Computer Science* **1017** 88–100.

Rosen, B. K. (1975) Deriving graphs from graphs by applying a production. *Acta Informatica* **4** 337–357.

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

Schürr, A., Winter, A. and Zündorf, A. (1999) PROGRES: Language and Environment. In: Ehrig, H., Engels, G., Kreowski, H.-J. and Rozenberg, G. (eds.) *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages, and Tools*, World Scientific 487–550.

Uesu, T. (1978) A system of graph grammars which generates all recursively enumerable sets of labelled graphs. *Tsukuba Journal of Mathematics* **2** 11–26.

https://doi.org/10.1017/S0960129501003425 Published online by Cambridge University Press