Parallel and distributed derivations in the single-pushout approach*

H. Ehrig and M. Löwe
FR 6-1, Technische Universität Berlin, Franklinstrasse 28/29, W-1000 Berlin 10, Germany

Abstract

Parallel and distributed derivations are introduced and studied in the single-pushout approach, which models rewriting by pushout constructions in appropriate categories of partial morphisms. We present a categorical framework for this approach in an axiomatic way. Models of this categorical framework are among others: graphs, hypergraphs, relational structures, and algebraic specifications with suitable partial morphisms. Several new results concerning parallelism and distributed parallelism are presented which are even new in the example categories.

1. Introduction

Graph grammars have been used to specify various kinds of database and software systems, where the graphs correspond to the states and the graph productions to the operations or transformations of the system. Concepts of parallel and distributed productions and derivations in the algebraic approach are very useful to model concurrent access, aspects of synchronization, and distributed state graphs (see [5, 6, 12, 16-18]). Distributed systems based on global states have been studied in [3, 4].

A distributed state in our framework is modeled by a family of local state graphs together with a specification of the shared subgraph for each pair of local-state graphs $G_1$ and $G_2$. This specification consists of an interface graph $I$ and two embedding morphisms $I \rightarrow G_1$ and $I \rightarrow G_2$ which indicate the interface part in $G_1$ and $G_2$. The interface graph $I$ together with the two embeddings is called the interface of $G_1$ and

*Correspondence to: Prof. Dr. H. Ehrig, FR 6-1, Technische Universität Berlin, Franklinstrasse 28/29, W-1000 Berlin 10, Germany

* This work has partly been supported by the ESPRIT Basic Research Working Group No 3799 “Computing by Graph Transformation”.

0304-3975/93/$06.00 \copyright$ 1993—Elsevier Science Publishers B.V. All rights reserved
The global state of the system can be constructed by gluing together all the local state graphs along the corresponding interfaces, i.e. by constructing the colimit of the distributed state. If $G$ is the global state for a distributed state $D$, we call $D$ a splitting or distribution of $G$.

The main concepts and results for parallel and distributed derivations in the algebraic approach to graph grammars (double-pushout approach) are given in [6]. The distributed parallelism theorem in [6] states the equivalence of the notions of parallel-independent direct derivations, direct parallel derivations and strict direct distributed derivations. Direct derivations of the same global state are parallel-independent if they can be performed in arbitrary sequential order and lead to the same final state. Direct parallel derivations are simply direct derivations with the parallel rule constructed from a set of given rules by componentwise disjoint union. Strict distributed derivations consist of a direct derivation for each local state which preserves all interfaces.

The equivalence of these concepts can be generalized to high-level replacement systems as studied in [7] using the notion of HLR1-categories. This concept includes high-level replacement systems based on graphs, hypergraphs, structures, algebraic specifications and Petri nets. The parallelism theorem in [7] states the equivalence of the first two concepts. But the proof of the distributed parallelism theorem in [6] for the graph case can be generalized to HLR1-categories showing the equivalence of all three concepts for HLR1-systems based on the double-pushout approach.

In the single-pushout approach, which models transformation rules by partial morphisms and direct derivations by pushouts in categories of partial morphisms, it has already been shown that parallel independence implies parallel derivations but not vice versa (see [21]) unless we require specific additional requirements for the redices corresponding to parts of the gluing condition in the double-pushout approach (see [19]). A good introduction to single-pushout graph rewriting is presented in [20].

In Section 2 of this paper we present a simple categorical framework for the single-pushout approach which can be seen as counterpart for HLR1-categories in [7]. In Section 3, a characterization for parallel derivations in terms of direct asynchronous parallel transformations and a special kind of distributed derivations (parallel derivation theorem; cf. Theorem 3.4) is presented within this purely categorical framework. In the hierarchy theorem for distributed derivations (cf. Theorem 3.6) we show that there are proper implications between the following concepts:

1. strict direct I-distributed derivations with total splittings,
2. parallel-independent direct derivations,
3. direct parallel derivations, and
4. strict direct I-distributed derivations with partial splittings.

A partial splitting for a global state graph admits partial morphisms as interface embeddings, while total splittings restrict the interface embeddings to total morphisms.
While all distributed derivations in Section 3 are strict, i.e. preserve the interfaces, Section 4 addresses the case where the interfaces are affected by distributed derivations, so-called \textit{distributed derivations with dynamic interfaces}. The dynamic distributed derivation theorem (Theorem 4.3) shows that direct dynamic distributed derivations are equivalent to direct derivations on global states with corresponding amalgamated rules. \textit{Amalgamated rules} generalize the concept of parallel rules (cf. [1] for the double-pushout approach). Amalgamated rules can be constructed for rules which share common subrules. The amalgamation construction takes care that all rules are glued together such that they exactly overlap in the specified subrules. The construction of parallel rules, i.e. disjoint union, is amalgamation w.r.t. “empty subrules”. The proof of this theorem is based on the 4-cube-lemma presented in [12].

The categorical framework for the single-pushout approach in this paper is different from that in [15] and includes graph structures in the sense of [19] as well as relational structures and specifications with strict partial morphisms, compare Examples 2.4 and 2.6 and Lemma 2.5.

We presuppose that the reader is familiar with the basic notions of category theory, especially w.r.t. colimit constructions, as they are introduced in [13] or in the appendix of [11]. For most of the constructions and results, examples are given based on a simplified graph grammar model of a distributed police database.

\section{Categorical framework for single-pushout derivations}

Single-pushout graph transformations on graphs have been introduced and investigated, for example, in [23, 14, 15, 2]. The algebraic approach to graph transformation has been adapted to single-pushout transformations in [21]. The classical algebraic approach, which is based on a gluing construction for graphs modeled by a double-pushout diagram of \textit{total} morphisms, has turned out to be an important but special case of the single-pushout approach, in which the gluing of graphs is modeled by a single-pushout diagram in the more complex category of \textit{partial} morphisms.

A more general framework for the single-pushout derivations of \textit{algebraic structures}, so-called graph structures, is presented in [19, 20], which allows a uniform treatment of unlabeled and labeled graphs, hypergraphs, and other graph-like structures.

Here we generalize from concrete approaches and try to summarize the essence of this type of transformations on a purely categorical level. This leads to an approach that is, on the one hand, applicable to an even richer class of models including relational structures and algebraic specifications; compare Examples 2.4 and 2.6. On the other hand, the categorical form of the approach provides short and clear proofs of statements about parallel and distributed derivations which are long and tedious to prove on the level of concrete objects. The ideas, but not the technical details, are similar to the concept of high-level-replacement system in [7].
General assumption. 2.1. In this paper we assume to have a category $\text{CAT}$ with pushouts and a subcategory $\text{CAT}'$ of $\text{CAT}$ with initial object and pushouts which are preserved by the inclusion functor $I: \text{CAT}' \to \text{CAT}$.

According to our standard examples (Examples 2.2–2.4 and 2.6), we call the morphisms in $\text{CAT}$ partial morphisms and those in $\text{CAT}'$ total morphisms, although other interpretations of the morphisms are possible (see Example 2.7).

Especially we can conclude from our assumptions that $\text{CAT}'$ has also binary coproducts $A + B$, with total coproduct inclusions $A \to A + B$ and $B \to A + B$, which is also coproduct in $\text{CAT}$, and for morphisms $f: A \to C$ and $g: B \to C$ the induced morphism $(f, g): A + B \to C$ is total if and only if $f$ and $g$ are total. There are many instances of this general framework as the following examples demonstrate.

Example 2.2 (Sets with partial functions). Let $\text{CAT}$ be the category of sets with partial functions and $\text{CAT}'$ the category of sets with total functions. For constructions of pushouts in the category $\text{CAT}'$ compare [23, 9, 20].

Example 2.3 (Graph structures with partial Sig-homomorphisms). Graph structures are exhaustively studied in [20]. Therefore, we only present the essential idea how these structures can be seen in the categorical framework of General assumption 2.1: Graph structures are algebraic signatures $\text{Sig} = (S, \text{OP})$ where all operation symbols $N \in \text{OP}$ are unary, i.e. of the form $N : s1 \to s2$. Let $\text{CAT}'$ be the category of all $\text{Sig}$-algebras over a graph structure $\text{Sig}$ and all (total) $\text{Sig}$-homomorphisms. A partial $\text{Sig}$-homomorphism $f: A \to B$ consists of a $\text{Sig}$-subalgebra $A'$ of $A$ and a total $\text{Sig}$-homomorphism $f': A' \to B$. $\text{CAT}$ is the category of all $\text{Sig}$-algebras with partial $\text{Sig}$-homomorphisms. Standard examples are graphs and hypergraphs with partial and total morphisms (see [19, 20] for more details including the fact that $\text{CAT}$ and $\text{CAT}'$ satisfy General assumption 2.1).

Example 2.4 (Structures with strict partial morphisms). Structures $G = (G_\text{A}, G_\text{F})$ consist of a set of atoms $G_\text{A}$ and a set of formulas $G_\text{F}$ of the form $P(x_1, \ldots, x_n)$ over a given set $P$ of $n$-ary predicates such that $P \in P$ and $x_1, \ldots, x_n \in G_\text{A}$. Let $\text{CAT}'$ be the category of structures with (total) structure morphisms (compare [8], where the morphisms are defined in a way such that each morphism is uniquely determined by its assignment of atoms) and $\text{CAT}$ the category of structures with strict partial structure morphisms $f: G_1 \to G_2$, i.e. the domain $G_1'$ of $f$ is a strict inclusion $G_1' \subseteq G_1$ in the sense that $G_1'_A \subseteq G_1_A$, $G_1'_F = G_1_F \cap G_1_A^*$, where $G_1_A^*$ is the set of all formulas over the atoms $G_1'_A$ of $G_1'$.

Lemma 2.5 (Pushouts of strict partial structure morphisms). The category of all structures and strict partial structure morphisms in the sense of Example 2.4 has all pushouts. They can be constructed by a unique extension of the pushouts for partial functions on the $A$-component.
Proof. Given strict partial structure morphisms \( f : G \rightarrow H \) and \( g : G \rightarrow K \), construct \( (S_A, f_A^* : K_A \rightarrow S_A, g_A^* : H_A \rightarrow S_A) \) as the pushout of the partial functions \( f_A \) and \( g_A \) (for this construction compare \([23, 20]\)). Now define \( S_F \) as the set of formulas containing:

1. \( P(f_A^*(x_1), \ldots, f_A^*(x_n)) \) if \( P(x_1, \ldots, x_n) \in K_F \) and \( f_A^* \) defined for \( x_i \) with \( i = 1, \ldots, n \).
2. \( P(g_A^*(x_1), \ldots, g_A^*(x_n)) \) if \( P(x_1, \ldots, x_n) \in H_F \) and \( g_A^* \) defined for \( x_i \) with \( i = 1, \ldots, n \).

By definition of \( S_F, f_A^* \) and \( g_A^* \) can be extended to strict partial structure morphisms \( f^* : K \rightarrow S \) and \( g^* : H \rightarrow S \). Note that each strict partial morphism \( f : A \rightarrow B \) is uniquely determined by its atom component! Hence, we have \( f^* \circ g = g^* \circ f \) since \( f_A^* \circ g_A = g_A^* \circ f_A \) due to the pushout property of \( f_A^* \) and \( g_A^* \). This property also provides a unique \( u_A : S_A \rightarrow E \) with \( u_A \circ f_A^* = f_A^* \) and \( u_A \circ g_A^* = g_A^* \) for each pair of strict partial morphisms \( f : K \rightarrow E \) and \( g : H \rightarrow E \) that satisfy \( f \circ g = g \circ f \). It remains to show that \( u_A \) is extendible to the formulas in \( S_F \), i.e. \( P(s_1, \ldots, s_n) \in S_F \) and \( u_A \) defined for \( s_i \) with \( i = 1, \ldots, n \), then \( P(u_A(s_1), \ldots, u_A(s_n)) \in E_F \). Since \( S_F \) is defined by (1) and (2) above, we assume without loss of generality for \( i = 1, \ldots, n : s_i = f_A^*(k_i) \) and \( P(k_1, \ldots, k_n) \in K_F \). With \( u_A \circ f_A^* = f_A^* \) \( f_A^* \) must be defined for \( k_1, \ldots, k_n \) which implies by strictness of \( f \) that \( f_A^* \) is defined for \( P(k_1, \ldots, k_n) \) and \( f_A^*(P(k_1, \ldots, k_n)) = P(f_A^*(k_1), \ldots, f_A^*(k_n)) = P(u_A \circ f_A^*(k_1), \ldots, u_A \circ f_A^*(k_n)) = P(u_A(s_1), \ldots, u_A(s_n)). \) Hence, \( u_A \) can be extended to a structure morphism \( u \).

Example 2.6 (Algebraic specifications with strict partial morphisms). Let \( \text{CAT}^* \) be the category \( \text{CATSPEC} \) of algebraic specifications \( \text{SPEC} = (S, \text{OP}, E) \) and (total) specification morphisms (see \([10]\)) and \( \text{CAT} \) the category of corresponding strict partial morphisms \( f : \text{SPEC}1 \rightarrow \text{SPEC}2 \), i.e. the domain of \( f \) is a strict inclusion \( \text{SPEC}1 \subseteq \text{SPEC}1 \) in the sense that \((S1', \text{OP}1') \subseteq (S1, \text{OP}1) \) and \( E1' = E1 \cap \text{Eqns}(S1', \text{OP}1') \), where \( \text{Eqns}(S1', \text{OP}1') \) is the set of all equations over \((S1', \text{OP}1')\). The pushout construction in \( \text{CAT} \) is similar to that in Example 2.4 using Example 2.3 for the construction on the signature component viewed as a hypergraph and unique extensions for the E-component.

Example 2.7 (Sets and finite sets). Take \( \text{CAT} \) to be the category of all sets and functions and \( \text{CAT}^* \) to be the full subcategory of all finite sets. This example shows that \( \text{CAT}^* \) is not necessarily a category of “partial morphisms” over \( \text{CAT} \) as in all examples above.

Using the categorical framework of General assumption 2.1, the basic concepts of the single-pushout approach are the following:

Definition 2.8 (Basic concepts of the single-pushout approach).

- A rule \( r : L \rightarrow R \) is a partial morphism.
- A redex of a rule \( r : L \rightarrow R \) in some object \( G \) is a total morphism \( m : L \rightarrow G \).
- The application of a rule \( r : L \rightarrow R \) at a redex \( m : L \rightarrow G \) leads to a direct derivation \((r, m) : G \Rightarrow H \) (short \( r : G \Rightarrow H \)) given by Diagram 1, which is a pushout, in \( \text{CAT} \) (of partial morphisms).
Remark. The morphism \( r' : G \to H \) is called \textit{direct derivation morphism} of \((r, m) : G \Rightarrow H\). In what follows, we represent a direct derivation \((r, m) : G \Rightarrow H\) by this derivation morphism if the applied rule \(r\) and the used redex \(m\) are obvious from the context.

Note that arbitrary derivations \(G_1 \Rightarrow^* G_n\) can easily be defined as sequences of direct derivations. Again mother graph and derived graph are connected by a \textit{derivation morphism}, i.e. the composition of the direct derivation morphisms for the direct derivations. In this article, all results are proven for direct derivations only but all definitions in the sequel can be easily extended to derivation sequences if direct derivation morphisms are substituted by arbitrary derivation morphisms.

3. Parallel and distributed derivations

Analogously to the transformation approach based on double-pushout construction [5], the basic concepts of Definition 2.8 can be easily enriched by notions of parallel and distributed derivations.

**Definition 3.1 (Parallel rules and derivations).**

- The \textit{parallel rule} of \( r_1 : L_1 \to R_1 \) and \( r_2 : L_2 \to R_2 \) is given by \( r_1 + r_2 : L_1 + L_2 \to R_1 + R_2 \), where \(+\) is the binary coproduct.
- A (direct) \textit{parallel derivation} \( r_1 + r_2 : G \Rightarrow G' \) is a direct derivation with a parallel rule \( r_1 + r_2 \).
- Direct derivations \( r_1 : G \Rightarrow H_1 \) and \( r_2 : G \Rightarrow H_2 \) are called \textit{parallel-independent} if in Diagram 2 with pushouts (1)–(3), the morphisms \( r_2 \circ m_1 : L_1 \to H_2 \) and \( r_1 \circ m_2 : L_2 \to H_1 \) are total.
- A \textit{direct asynchronous parallel derivation} \( r_1 \parallel r_2 : G \Rightarrow G' \) is given by direct derivations \( r_1 : G \Rightarrow H_1 \), \( r_2 : G \Rightarrow H_2 \) and \( G' \) given by the pushout (3) in Diagram 2.

**Remarks.** Asynchronous derivations are a tool to describe the joint effect of two direct derivations from the same graph \( G \) in general. This is due to the fact that this type of derivation is also defined for parallel-dependent direct derivations. Parallel and asynchronous derivations transform objects which represent a (global) state in some system. By contrast, distributed derivations presuppose that the global state has internal structure, i.e. is a collection of \( n \) local states which can be transformed separately. The investigation in this article is restricted to \( n = 2 \).
Definition 3.2 (Splitting and distributed derivations).

- A splitting of an object $G$ into objects $G_1$ and $G_2$ with interface $I$, written $G = G_1 +_I G_2$, is given by a pushout as it is depicted by (1) in Diagram 3. If all morphisms in this pushout are total, the splitting is called total, otherwise the splitting is called partial. If a splitting situation as in Diagram 3 is given, the object $G$ is referred to as the global state and the pair of morphisms $(I \rightarrow G_1, I \rightarrow G_2)$ represents the corresponding $I$-distributed state of $G$.

- A direct $I$-distributed derivation with total (partial) splittings $G = G_1 +_I G_2$ and $G' = G_1' +_I G_2'$, defined by total (partial) pushouts (1) and (1') in Diagram 4, is given by (local) direct derivations $G_1 \ast G_1'$, $G_2 = G_2'$ with direct derivation morphisms $d_1 : G_1 \rightarrow G_1'$, $d_2 : G_2 \rightarrow G_2'$ such that Diagram 4 commutes, i.e. $d_1 \circ i_1 = i_2$ and $d_2 \circ i_3 = i_4$.

- The splitting $G = G_1 +_I G_2$ with morphisms $g_i : G_i \rightarrow G$ ($i = 1, 2$) is called compatible with global redices $m_1 : L_1 \rightarrow G$ and $m_2 : L_2 \rightarrow G$ if there are redices $n_1 : L_1 \rightarrow G_1$ and $n_2 : L_2 \rightarrow G_2$ s.t. $g_1 \circ n_1 = m_1$ and $g_2 \circ n_2 = m_2$, which is indicated by the $=$ symbol in Diagrams 5 below.

- The splitting $G = G_1 +_I G_2$ with $g_1, g_2$ as above is called compatible with local redices $n_1 : L_1 \rightarrow G_1$, $n_2 : L_2 \rightarrow G_2$, if $m_1$ and $m_2$ (as defined above) are total.

Remarks. Note that a splitting which consists of total morphisms only, is always compatible with local redices. This is not true for partial splitting since the pushout of
partial morphisms leads to partial embeddings of the local state into the resulting global state. Thus, the composition of local reductions with the embeddings of local into global states need not be total, which means that they need not be reductions in the global state; compare Definition 2.8.

The notion of direct distributed derivation as it is introduced above can easily be extended to distributed derivation sequences if the direct derivation morphisms $d_1$ and $d_2$ are substituted by arbitrary derivation morphisms for whole sequences, compare remark after Definition 2.8.

**Example 3.3 (Police database).** The main concept of splitting and distributed derivation is explained below by a small example which resumes the graph grammar model of a police database [22] presented in [20]. It uses simple labeled graphs as the underlying category (compare [20] for pushout constructions). The database has two sets of data, personal data (white) and case data (black). The sets are represented by big vertices, the elements by small vertices. Figure 1 visualizes the graph grammar rules for object creation, rules (1) and (2), and relation insertion, rule (3). We allow also rules analogous to (3) which insert relations among persons only (cases only). By inverting these rules, we obtain a description for object and relation deletion. If the rule morphisms are only indicated by the symbol $\Rightarrow$, we implicitly assume that the rule morphisms map all objects of the rule's left-hand side injectively to objects in the right-hand side which have the same graphical layout left and right of $\Rightarrow$. We use the $\Rightarrow$ to represent morphisms in the example because simple arrows are already used to indicated edges in the graphs. Figure 2 shows a parallel rule built from the rules “delete person p” and “insert father relation between $q$ and $r$”. Figure 3 visualizes a total splitting of a database $G$. There is one relation crossing the border between the components $G_1$ and $G_2$, which much be put in the interface. It cannot be worked on locally by the rules (given above) since $q$ and $c$ do not belong to the set of person data represented in $G_1$ and to the set of case data represented in $G_2$, respectively.
Parallel and distributed derivations in the single-pushout approach are closely related as the first main theorem demonstrates.

**Theorem 3.4 (Parallel derivation).** Given rules \( r_1 : L_1 \rightarrow R_1 \) and \( r_2 : L_2 \rightarrow R_2 \) the following three statements are equivalent:

1. There is a direct parallel derivation \( r_1 + r_2 : G \Rightarrow G' \) with redex \((m_1, m_2) : L_1 + L_2 \rightarrow G\) and a partial splitting \( G = G +_I G_2 \) compatible with the global redices \( m_1 : L_1 \rightarrow G \) and \( m_2 : L_2 \rightarrow G \).

2. There is a direct asynchronous derivation \( r_1 || r_2 : G \Rightarrow G' \) with direct derivations \( (r_1, m_1) : G \Rightarrow H_1 \) and \( (r_2, m_2) : G \Rightarrow H_2 \) and a partial splitting of \( G \) compatible with the global redices \( m_1 \) and \( m_2 \).

3. There is a direct \( I \)-distributed derivation with partial splittings \( G = G_1 +_I G_2 \), \( G' = G_1' +_I G_2' \) and (local) direct derivations \( (r_1, n_1) : G \Rightarrow G_1' \) and \( (r_2, n_2) : G \Rightarrow G_2' \) such that the splitting \( G = G_1 +_I G_2 \) is compatible with the local redices \( n_1 : L_1 \rightarrow G_1 \) and \( n_2 : L_2 \rightarrow G_2 \).

**Proof.** The equivalence of (1) and (2) is given by the butterfly lemma (see [7]), which states the equivalence of pushout (0) with pushouts (1)–(3) shown in Diagram 6. Moreover, \( (m_1, m_2) \) is total iff \( m_1 \) and \( m_2 \) are total (see General assumption 2.1).

The equivalence of (2) and (3) follows from the following staircase diagram (Diagram 7), where (4) corresponds to the partial splitting of \( G \) compatible with \( m_1 \) and \( m_2 \). Given statement (2), the partial splitting of \( G \) compatible with \( m_1 \) and \( m_2 \) leads to pushout (4) with \( m_i = g_i \circ n_i \), where \( n_i \) is total for \( i = 1, 2 \). Moreover, we obtain pushouts (5) and (6) ((7) and (8)) as decomposition of pushout (1) ((2)). The composition of pushouts (4), (6), (8), and (3) leads to the splitting \( G = G_1' +_I G_2' \) such that statement (3) is satisfied. Vice versa, given statement (3), we have pushouts (4), (5), (7), and Diagram 8, with \( I \rightarrow G_i' = I \rightarrow G_i \rightarrow G_i' \) and total morphisms \( m_i = g_i \circ n_i \) for \( i = 1, 2 \) by assumption. This allows to decompose pushout (9) into pushouts (4), (6), (8) and (3) such that statement (2) is satisfied defining pushouts (1) ((2)) as composition of pushouts (5) and (6) ((7) and (8)). \(\Box\)

**Example 3.5 (Local update of distributed system).** The power of distributed derivations, especially their potential to perform global update in the system by local activities, is demonstrated by the following distributed derivation on the sample splitting of Fig. 3. If we apply a rule for the deletion of \( q \) in \( G_2 \) and the rule which adds case \( b \) in \( G_1 \), the interface embedding of \( I \) into \( G_2' \) becomes partial (cf. Fig. 4). Hence,
Fig. 3. Distribution of a sample database.
the pushout construction (depicted in Fig. 4) which calculates the next global state, erases vertex $q$ with all incident edges from $G_1'$ as well (cf. [20] for details of pushouts of partial graph morphisms).

A criterium that local activities have local effect seems to be that the interfaces embeddings remain total. That this is actually a synchronization requirement is shown by the following hierarchy theorem.
Fig. 4. Local updates of distributed systems.
Theorem 3.6 (Hierarchy for distributed derivations). Given rules $r_1 : L_1 \to R_1$ and $r_2 : L_2 \to R_2$ we have, for $n = 1, 2, 3$, that statement $n$ implies statement $n + 1$ and none of the implications is an equivalence:

1. There is a direct $I$-distributed derivation with total splittings $G = G_1 +_I G_2$, $G' = G_1' +_I G_2'$ and local direct derivations $(r_1, n_1) : G_1 \Rightarrow G_1'$ and $(r_2, n_2) : G_2 \Rightarrow G_2'$.

2. There are parallel-independent direct derivations $(r_1, m_1) : G \Rightarrow H_1$ and $(r_2, m_2) : G \Rightarrow H_2$ and a total splitting $G = G_1 +_I G_2$ compatible with the global redices $m_1$ and $m_2$.

3. There is a direct parallel derivation $r_1 + r_2 : G \Rightarrow G'$ with redex $k_1, k_2 : L_1 + L_2 \to G$ and a partial splitting $G = G_1 +_I G_2$ compatible with the global redices $m_1 : L_1 \to G$ and $m_2 : L_2 \to G$.

4. There is a direct $I$-distributed derivation with partial splittings $G = G_1 +_I G_2$, $G' = G_1' +_I G_2'$ and local direct derivations $r_1 : G_1 \Rightarrow G_1'$ and $r_2 : G_2 \Rightarrow G_2'$.

Remark. Note that statement (3) coincides with statement (1) of Theorem 3.4 and we obtain statement (4) from statement (1) if total splittings are replaced by partial splittings.

Proof. We consider Diagram 7, which can be obtained from each statement similar to the proof of Theorem 3.4. We only have to check which of the morphisms are total.

Given the first statement $n_1, n_2$ and the diagrams (4) and (9) as composition of (4), (6), (8) and (3), are total. This means that $I \to G_1'$ and $I \to G_2'$ are total and, hence, also $G_2 \to H_1$ and $G_1 \to H_2$ are total because (4) + (6) and (4) + (8) are total pushouts [but not necessarily (6) and (8)]. This implies that $L_2 \to G_2 \to H_1$ and $L_1 \to G_1 \to H_2$ are total implying statement (2).

Statement (2) implies Theorem 3.4(2) since total splittings are special partial splittings. But Theorem 3.4(2) is equivalent to Theorem 3.4(1) which implies our statement (3).

Statement (3) is a specialization of Theorem 3.4(1). This is equivalent to Theorem 3.4(3) which implies our statement (4).

The database example can be used to construct counterexamples showing that none of the implications is an equivalence.

For a situation satisfying statement (4) and not statement (3), consider a slight modification of the splitting in Fig. 3: Add a person $t$ in $G_2$ and $I$ together with an unlabeled edge from $t$ to the personal database vertex in $G_2$. Now the embedding of $I$ into $G_1$ is partial while $G$ is also the pushout of the modified situation. Now we can locally delete this extra $t$ in $G_2$ and $p$ in $G_1$. The corresponding parallel rule is not applicable to $G$ since there is no $t$-labeled vertex. The example demonstrates that the splittings and interfaces must be carefully chosen for consistent local behavior. Splittings generated as described in Example 3.3 prohibit these effects in the police database example.

For a situation satisfying statement (3) and not statement (2), consider Fig. 5, which shows a direct transformation with the parallel rule of Fig. 2 at a noninjective redex.
The redices of the component rules are not parallel-independent since they overlap in the objects 6 and 7 as well as in 4 and 5 (see [20]).

The last counterexample has already been described in Example 3.5. If we apply two rules deleting \( p \) or \( q \) to the global state \( G \) in Fig. 3, the resulting transformations are parallel-independent and the splitting in Fig. 3 is compatible with the global redices. Nevertheless, the resulting splitting of the local transformations is partial. □

4. Distributed derivations with dynamic interfaces

In this section we introduce distributed derivations with dynamic interfaces, which means that the two distributed derivations induce a common derivation \( r0 : I \leadsto I' \) on the interface. The dynamic distributed derivation theorem shows that direct \( (I, I') \)-distributed derivations are equivalent to direct global derivations with corresponding amalgamated rules. (Amalgamation is a concept introduced in [1, 20], which is able to describe handshake synchronization of rules in distributed systems; cf. [4].)

**Definition 4.1 (Dynamic distributed derivations).** A direct \( (I, I') \)-distributed derivation (or short dynamic distributed derivation) with total (partial) splittings \( G = G1 +_r G2 \) and \( G' = G1' +_r G2' \), defined by total (partial) pushouts (1) and (1') as in Diagram 9, is given by local direct derivations \( G1 \Rightarrow G1' \), \( G2 \Rightarrow G2' \), and \( I \Rightarrow I' \) with derivation morphisms \( d1 : G1 \to G1' \), \( d2 : G2 \to G2' \), and \( d0 : I \to I' \) such that Diagram 9 commutes.

**Remarks.** Note that the situation in a dynamic distributed derivation depicted in Diagram 9 can be obtained from the situation of (static) distributed derivation (cf.
Diagram 4) by inserting an additional derivation morphism between the interfaces. Conversely, Diagram 4 can be obtained from the diagram above by letting $d_0 = \text{id}_I$.

Dynamic distributed derivations allow local derivations which change the interface of a distributed state to take place if both derivations induce the same "derivation" of the interface. Thus, the local transformations are "synchronized" due to the requirement that the interface must be updated consistently, i.e. in the same way.

If the associated global states are considered, we note that any sequential transformation with the same (local) rules leads to a different result (in general) since the synchronization effect is lost. It can be reestablished if so-called amalgamated rules and direct transformations with them on global states are considered. Amalgamated rules are in some sense "synchronized rules" w.r.t. a shared subrule.

General $(I, I')$-distributed derivations can be obtained from the notion above if the morphisms $d_0, d_1,$ and $d_2$ are supposed to be derivation morphisms of derivation sequences.

**Definition 4.2 (Amalgamated Rules).**

- A rule $r_0 : L_0 \rightarrow R_0$ is called a subrule of rule $r_1 : L_1 \rightarrow R_1$ defined by total morphisms $L_0 \rightarrow L_1$ and $R_0 \rightarrow R_1$ if Diagram 10 commutes.
- Given rules $r_1 : L_1 \rightarrow R_1, r_2 : L_2 \rightarrow R_2$ with common subrule $r_0 : L_0 \rightarrow R_0$ defined by total morphisms $L_0 \rightarrow L_i$ and $R_0 \rightarrow R_i$ for $i = 1, 2$ the amalgamated rule $r_3 = r_1 + r_0 r_2 : L_3 \rightarrow R_3$ is defined by the pushouts (1) and (2) and the induced morphism $r_3 : L_3 \rightarrow R_3$ in the following 3-cube (Diagram 11).
- A total splitting $G = G_1 + G_2$ given by pushout (3) is called compatible with the amalgamated rule $r_3 = r_1 + r_0 r_2$ and a redex $m_3 : L_3 \rightarrow G$ if there are redices $m_i : L_i \rightarrow G_i$ for $i = 0, 1, 2$ with $G_0 = I$ such that the following 3-cube (Diagram 12) commutes.

![Diagram 10]
Dynamic distributed derivations and derivations with amalgamated rules are closely related as the following theorem shows.

**Theorem 4.3** (Direct dynamic distributed derivation). Given rules $r_i: L_i \rightarrow R_i$ for $i = 0, 1, 2$ such that the amalgamated rule $r_3 = r_1 + r_2$ is defined, the following statements are equivalent:

1. There is a direct amalgamated derivation $r_3: G \Rightarrow G'$ with redex $m_3: L_3 \rightarrow G$ and a total splitting $G = G_1 + G_2$ which is compatible with the amalgamated rule $r_3$ and the redex $m_3$.

2. There is a direct $(I, I')$-distributed derivation with total splitting $G = G_1 + G_2$ and partial splitting $G' = G_1' + G_2'$ given by local direct derivations $r_0: I \Rightarrow I'$, $r_1: G_1 \Rightarrow G_1'$ and $r_2: G_2 \Rightarrow G_2'$.

**Remark.** Statements (1) and (2) can be combined in Diagram 13, where $r_0'$, $r_1'$, $r_2'$, and $r_3'$ are the derivation morphisms induced by the direct derivation with the rules $r_0 \rightarrow r_3$. In general, the result $G'$ of the direct $(I, I')$-distributed derivation does not coincide with the result of the corresponding parallel derivation with the rule $r_1 + r_2$.

**Proof.** In order to show the equivalence of both statements, we construct in both cases the commutative 4-cube depicted in Diagram 14, given by one 3-cube within another.
3-cube. The 4-cube-lemma presented in [12] considers two sequences of parallel squares, e.g.

(1) \(R0 \rightarrow R1 \rightarrow R2 \rightarrow R3\) (5) \(L0 \rightarrow L0 \rightarrow L\) (6) \(L1 \rightarrow R1 \rightarrow G1 \rightarrow G1'\)

(2) \(L1 \rightarrow L2 \rightarrow L3\) (7) \(L2 \rightarrow L2 \rightarrow R2 \rightarrow G2 \rightarrow G2'\)

(3) \(I \rightarrow G1 \rightarrow G2 \rightarrow G\) (8) \(L3 \rightarrow L3 \rightarrow R3 \rightarrow G \rightarrow G'\)

(4) \(I' \rightarrow G1' \rightarrow G2' \rightarrow G'\)

where the intersection of the squares (4) and (8) consists exactly of the object \(G'\): If the squares (1)–(3) and (5)–(7) are pushouts then (4) is a pushout iff (8) is a pushout. This 4-cube-lemma is valid in any category and can be derived as a special case of commutativity of colimits (see [11, Appendix 10C]), where the given small categories \(S1\) and \(S2\) are both generated by pushout schemes and the scheme \(S1 \times S2\) corresponds to all objects in the 4-cube above except \(R3, L3, G, G', I', G1', G2'\) and all morphisms between the remaining objects.

Now we show the equivalence of statements (1) and (2).
Given statement (2), we have the inner 3-cube and top, left and back of the outer 3-cube in the 4-cube above, where the diagrams (1)–(3) are total and (4)–(7) are partial pushouts by assumption and $I' \rightarrow G1'$, $I' \rightarrow G2'$, $G \rightarrow G'$ and $R3 \rightarrow G'$ are induced morphisms using the pushout properties of (5), (3) and (1). Now the 4-cube-lemma implies that (8) also is a pushout which is the required direct amalgamated derivation $r3 : G \Rightarrow G'$. Note that $L3 \rightarrow G$ is total because $L1 \rightarrow G1$ and $L2 \rightarrow G2$ are total (redices of $r1 : G1 \Rightarrow G1'$ and $r2 : G2 \Rightarrow G2'$) and pushout (3) is total by assumption. Hence, we have statement (1), where the compatibility of the total splitting $G = G1 + G2$ with $r3$ and $m3$ corresponds to the commutativity of the inner 3-cube.

Conversely, given statement (1), we have the inner 3-cube with total morphisms and pushouts (2) and (3). Moreover, (1) and (8) are pushouts by assumptions. Now we construct the objects $I'$, $G1'$ and $G2'$ as pushout objects in (5), (6) and (7), respectively. Hence, we have local direct derivations $r0 : I \Rightarrow I'$, $r1 : G1 \Rightarrow G1'$ and $r2 : G2 \Rightarrow G2'$, where the redices are total morphisms because the inner 3-cube consists of total morphisms only. Finally, the 4-cube-lemma implies that also (4) is a pushout which leads to the required partial splitting $G' = G1' + G2'$ and, hence, to statement (2). □

Example 4.4 (Distributed derivation with dynamic interfaces). The concept of distributed derivations with dynamic interfaces offers a synchronization mechanism for “global changes” without the need for constructing the global state. Consider again the splitting in Fig. 3. The relation between $q$ and $c$ is global w.r.t. the components $G1$ and $G2$, i.e. it crosses the component border. With the rules proposed in the introduction, these global relations cannot be manipulated locally neither in $G1$ nor in $G2$. For

![Subrule](image1.png)

![Object](image2.png)

![Subject](image3.png)

Fig. 6. Rules for synchronized manipulation of global relations.
Fig. 7. Example of a result for a dynamic distributed derivation.
example, deletion of this relation requires to join both component graphs and to perform a global action.

Adding a local rule which allows to delete a relation whose subject or object is not part of the local component would do the job but it is unsatisfactory because the interface and the affected other local component remain unchanged. Thus, the global relation is not deleted in the whole system, which intuitively leads to inconsistent local states. This inconsistency can only be repaired by a gluing to a global state and a new splitting (compare situation in Fig. 3). Hence, what we need is the pair of rules in Fig. 6, which specify their joint global effect by a common subrule. In order to obtain an \((I, I')\)-distributed derivation the object and the subject rule of Fig. 6 performed in \(G1\) or \(G2\) can be synchronized at their common subrule, which specifies the effect on the interface. This kind of handshake operation for the police database example not only manipulates the interface without global state but also provides total embeddings of the resulting interface in the generated local components; cf. Fig. 7, which shows the result of the dynamic distributed derivation with the rules in Fig. 6 applied to the distributed situation of Fig. 3 (the subrule transforms \(I\) to \(I'\), the object rule transforms \(G1\) to \(G1'\), and the subject rule derives \(G2'\) from \(G2\)).

5. Conclusion

We have presented a general concept for distributed derivations in the single-pushout approach on the level of category theory. The concept is applicable to a variety of graph-like structures, like graphs and hypergraphs, and also to relational structures and algebraic specifications. The relationship of static and dynamic distributed derivations to parallel or amalgamated transformations on the associated global state are comprehensively studied above.

The most interesting feature of distribution as it is presented here is the difference between the expressive power of derivations with total or partial splittings. Future research shall focus on criteria for distributed derivation steps to preserve total splittings and the role which the extra information in the interface (in the case of partial splitting) can play in the design of distributed systems modeled by graph grammars. Furthermore, the whole theory shall be generalized from direct derivations to arbitrary derivation sequences.

Acknowledgments

We thank all the referees of the first draft version for their valuable and constructive comments. Last but not least, we thank Helga Barnewitz for excellent typing of the draft version.
References


