Technische Universität Berlin

Forschungsberichte der Fakultät IV – Elektrotechnik und Informatik

Multi-Amalgamation in $\mathcal{M}$-Adhesive Categories Long Version

Ulrike Golas

Bericht-Nr. 2010-05
ISSN 1436-9915

Multi-Amalgamation in $\mathcal{M}$-Adhesive Categories
Long Version

Ulrike Golas

Technische Universität Berlin, Germany
ugolas@cs.tu-berlin.de

Abstract. Amalgamation is a well-known concept for graph transformations in order to model synchronized parallelism of rules with shared subrules and corresponding transformations. This concept is especially important for an adequate formalization of the operational semantics of statecharts and other visual modeling languages, where typed attributed graphs are used for multiple rules with general application conditions. However, the theory of amalgamation for the double pushout approach has been developed up to now only on a set-theoretical basis for pairs of standard graph rules without any application conditions.
For this reason, we present the theory of amalgamation in this paper in the framework of $\mathcal{M}$-adhesive categories, short for weak adhesive HLR categories, for a bundle of rules with (nested) application conditions. The main result is the Multi-Amalgamation Theorem, which generalizes the well-known Parallelism and Amalgamation Theorems to the case of multiple synchronized parallelism.
The constructions are illustrated by a small running example. A more complex case study for the operational semantics of statecharts based on multi-amalgamation is presented in a separate paper.

1 Introduction

1.1 Historical Background of Amalgamation

The concepts of adhesive [1] and (weak) adhesive high-level replacement (HLR) [2] categories have been a break-through for the double pushout approach of algebraic graph transformations [3]. Almost all main results could be formulated and proven in these categorical frameworks and instantiated to a large variety of HLR systems, including different kinds of graph and Petri net transformation systems [2]. These main results include the Local Church–Rosser, Parallelism, and Concurrency Theorems, the Embedding and Extension Theorem, completeness of critical pairs, and the Local Confluence Theorem.

However, at least one main result is missing up to now. The Amalgamation Theorem in [4] has been developed only on a set-theoretical basis for a pair of standard graph rules without application conditions. In [4], the Parallelism Theorem of [5] is generalized to the Amalgamation Theorem, where the assumption of parallel independence is dropped and pure parallelism is generalized to synchronized parallelism. The synchronization of two rules p_1 and p_2 is expressed by a common subrule p_0, which we call *kernel rule* in this paper. The subrule concept is formalized by a rule morphism $s_i : p_0 \to p_i$, called *kernel morphism* in this paper, based on pullbacks and a pushout complement property. p_1 and p_2 can be glued along p_0 leading to an amalgamated rule $\tilde{p} = p_1 +_{p_0} p_2$. The Amalgamation Theorem states that each amalgamable pair of direct transformations $G \xRightarrow{p_i,m_i} G_i (i = 1, 2)$ via p_1 and p_2 leads to an amalgamated transformation $G \xRightarrow{\tilde{p},\tilde{m}} H$ via $\tilde{p}$, and vice versa yielding a bijective correspondence.

Moreover, the Complement Rule Theorem in [4] allows to construct a complement rule $\overline{p}$ of a kernel morphism $s : p_0 \to p$ leading to a concurrent rule $p_0 *_E \overline{p}$ which is equal to p. Now the Concurrency Theorem allows to decompose each amalgamated transformation $G \xRightarrow{\tilde{p},\tilde{m}} H$ into sequences $G \xRightarrow{p_i} G_i \xRightarrow{q_i} H$ for $i = 1, 2$ and vice versa, where q_i is the complement rule of $t_i : p_i \to \tilde{p}$.

The concepts of amalgamation are applied to communication based systems and visual languages in [4, 6, 7, 8, 9] and transferred to the single-pushout approach of graph transformation in [10].

1.2 The Aim of this Paper

The concept of amalgamation plays a key role in the application of parallel graph transformation to communication-based systems [8] and in the modeling of the operational semantics for visual languages [9]. However, in most of these applications we need amalgamation for n rules, called multi-amalgamation, based not only on standard graph rules, but on different kinds of typed and attributed graph rules including (nested) application conditions.

The main idea of this paper is to fill this gap between theory and applications. For this purpose, we have developed the theory of multi-amalgamation for adhesive and adhesive HLR systems based on rules with application conditions. This allows to instantiate the theory to a large variety of graphs and corresponding graph transformation systems and, using weak adhesive HLR categories, also to typed attributed graph transformation systems [2]. A complex case study for the operational semantics of statecharts based on typed attributed graphs and multi-amalgamation is presented in [11].

1.3 General Assumptions

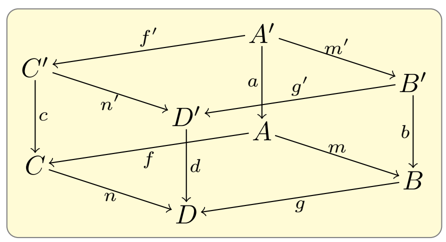
In this paper we assume to have an $\mathcal{M}$-adhesive category with binary coproducts, epi-$\mathcal{M}$-factorization, and initial pushouts [2]. We consider rules with (nested) application conditions [13] as explained below and assume that the reader is familiar with concurrent rules and the Concurrency Theorem. In the following, a *bundle* represents a family of morphisms or transformation steps with the same domain, which means that a bundle of things always starts at the same object.

1.4 Organization of this Paper

This paper is organized as follows. In Section 2, we review basic notions of $\mathcal{M}$-adhesive categories, transformations, and application conditions. In Section 3, we introduce kernel rules, multi rules, and kernel morphisms leading to the Complement Rule Theorem as first main result. In Section 4, we construct multi-amalgamated rules and transformations and show as second main result the Multi-Amalgamation Theorem. In Section 5, we present a summary of our results and discuss future work.

2 Review of Basic Notions

The basic idea of adhesive categories [1] is to have a category with pushouts along monomorphisms and pullbacks satisfying the van Kampen property. Intuitively, this means that pushouts along monomorphisms and pullbacks are compatible with each other. This holds for sets and various kinds of graphs (see [1, 2]), including the standard category of graphs which is used as a running example in this paper. $\mathcal{M}$-adhesive categories, called weak adhesive HLR category in [2], include a special morphism class $\mathcal{M}$ of monomorphisms and extend adhesive categories with suitable properties. As a main difference, they only require pushouts along $\mathcal{M}$-morphisms to be weak van Kampen squares.



Definition 1 (Van Kampen square). *A pushout as at the bottom of the cube on the right with $m \in \mathcal{M}$ is a* weak van Kampen square *if it satisfies the* weak van Kampen property*, i.e., for any commutative cube, where the back faces are pullbacks and ($f \in \mathcal{M}$ or $b, c, d \in \mathcal{M}$), the following statement holds: The top face is a pushout if and only if the front faces are pullbacks.*

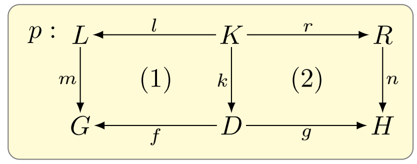
In contrast, the (non-weak) van Kampen property does not assume ($f \in \mathcal{M}$ or $b, c, d \in \mathcal{M}$).

Definition 2 ($\mathcal{M}$-adhesive category). *An* $\mathcal{M}$-adhesive category $(\mathbf{C}, \mathcal{M})$ *consists of a category* $\mathbf{C}$ *and a class $\mathcal{M}$ of monomorphisms in* $\mathbf{C}$*, which is closed under isomorphisms, composition, and decomposition ($g \circ f \in \mathcal{M}$ and $g \in \mathcal{M}$ implies $f \in \mathcal{M}$), such that* $\mathbf{C}$ *has pushouts and pullbacks along $\mathcal{M}$-morphisms, $\mathcal{M}$-morphisms are closed under pushouts and pullbacks, and pushouts along $\mathcal{M}$-morphisms are weak van Kampen squares.*

Well-known examples of $\mathcal{M}$-adhesive categories are the categories $(\mathbf{Sets}, \mathcal{M})$ of sets, $(\mathbf{Graphs}, \mathcal{M})$ of graphs, $(\mathbf{Graphs_{TG}}, \mathcal{M})$ of typed graphs, $(\mathbf{ElemNets}, \mathcal{M})$ of elementary Petri nets, $(\mathbf{PTNets}, \mathcal{M})$ of place/transition nets, where for all these categories $\mathcal{M}$ is the class of all monomorphisms, and $(\mathbf{AGraphs_{ATG}}, \mathcal{M})$ of typed attributed graphs, where $\mathcal{M}$ is the class of all injective typed attributed graph morphisms with isomorphic data type component (see [2]).

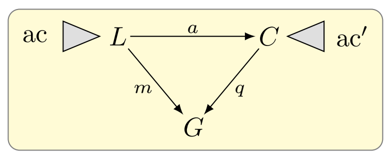
In the double pushout approach to transformations, rules describe in a general way how to transform objects. The application of a rule to an object is called a transformation and based on two gluing constructions, which are pushouts in the corresponding category.

Definition 3 (Rule and transformation). *A* rule *is given by a span* $p = (L \xleftarrow{l} K \xrightarrow{r} R)$ *with objects* L*,* K*, and* R*, called left-hand side, interface, and right-hand side, respectively, and* $\mathcal{M}$*-morphisms* l *and* r*. An application of such a rule to an object* G *via a match* $m : L \to G$ *is constructed as two pushouts* (1) *and* (2) *leading to a* direct transformation $G \xRightarrow{p,m} H$*.*



An important extension is the use of rules with suitable application conditions. These include positive application conditions of the form $\exists a$ for a morphism $a : L \to C$, demanding a certain structure in addition to L, and also negative application conditions $\neg\exists a$, forbidding such a structure. A match $m : L \to G$ satisfies $\exists a$ ($\neg\exists a$) if there is a (no) $\mathcal{M}$-morphism $q : C \to G$ satisfying $q \circ a = m$. In more detail, we use nested application conditions [13], short application conditions.

Definition 4 (Application condition and satisfaction). *An* application condition ac *over an object* L *is of the form* ac = true *or* $\text{ac} = \exists(a, \text{ac}')$*, where* $a : L \to C$ *is a morphism and* ac′ *is a condition over* C*.*



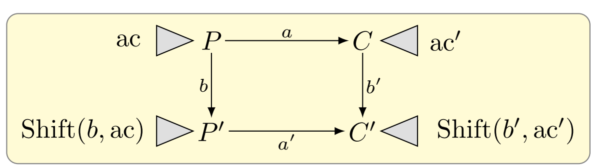
Given a condition ac *over* L*, then a morphism* $m : L \to G$ satisfies ac*, written* $m \models \text{ac}$*, if* ac = true *or* $\text{ac} = \exists(a, \text{ac}')$ *and there exists a morphism* $q \in \mathcal{M}$ *with* $q \circ a = m$ *and* $q \models \text{ac}'$*.*

Moreover, application conditions are closed under boolean formulas and satisfaction is extended as usual. For simplification, false *abbreviates* ¬true*,* $\exists a$ *abbreviates* $\exists(a, \text{true})$*, and* $\forall(a, \text{ac})$ *abbreviates* $\neg\exists(a, \neg\text{ac})$*. With* $\text{ac}_C \cong \text{ac}'_C$ *we denote the semantical equivalence of* ac_C *and* ac'_C *on* C*.*

In this paper we consider rules of the form $p = (L \xleftarrow{l} K \xrightarrow{r} R, \text{ac})$, where $(L \xleftarrow{l} K \xrightarrow{r} R)$ is a (plain) rule and ac is an application condition on L. In order to handle rules with application conditions there are two important concepts, called the shift of application conditions over morphisms and rules ([13, 14]).

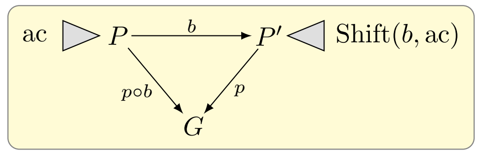
For the shift construction over morphisms, all epimorphic overlappings of the codomain of the shift morphism and the codomain of the condition morphism have to be collected.

Definition 5 (Shift over morphism). *Given an application condition* $\text{ac} = \exists(a, \text{ac}')$ *over* P *and a morphism* $b : P \to P'$*, then*



Shift(b, ac) *is an application condition over* P' *defined by* Shift$(b, \mathrm{ac}) = \vee_{(a',b') \in \mathcal{F}} \exists(a', \mathrm{Shift}(b', \mathrm{ac}'))$ *with* $\mathcal{F} = \{(a', b') \mid (a', b')$*jointly epimorphic*$, b' \in \mathcal{M}, b' \circ a = a' \circ b\}$. *Moreover,* Shift$(b, \mathrm{true}) = \mathrm{true}$ *and the construction is extended for boolean formulas in the usual way.*

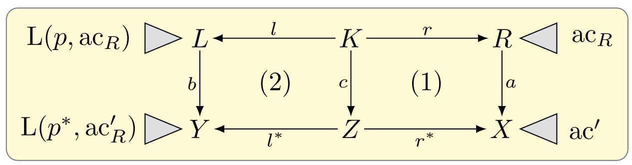
Fact 1. *Given an application condition* ac *over* P *and morphisms* $b : P \to P'$ *and* $p : P' \to G$*, then* $p \models \mathrm{Shift}(b, \mathrm{ac})$ *if and only if* $p \circ b \models \mathrm{ac}$.



Proof. See [13, 14]. □

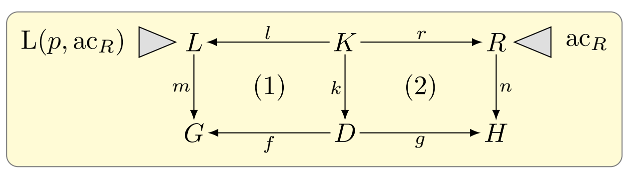
In analogy to the application condition over L, which is a pre application condition, it is also possible to define post application conditions over the right hand side R of a rule. Since these application conditions over R can be translated to equivalent application conditions over L (and vice versa) [13], we can restrict our rules to application conditions over L.

Definition 6 (Shift over rule). *Given a rule* $p = (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R, \mathrm{ac})$ *and an application condition* $\mathrm{ac}_R = \exists(a, \mathrm{ac}'_R)$ *over* R*, then* $\mathrm{L}(p, \mathrm{ac}_R)$ *is an application condition over* L *defined by* $\mathrm{L}(p, \mathrm{ac}_R) = \exists(b, \mathrm{L}(p^*, \mathrm{ac}'_R))$ *if* $a \circ r$ *has a pushout complement* (1) *and* $p^* = (Y \overset{l^*}{\leftarrow} Z \overset{r^*}{\rightarrow} X)$ *is the derived rule by constructing pushout* (2)*, otherwise* false. *Moreover,* $\mathrm{L}(p, \mathrm{true}) = \mathrm{true}$ *and the construction is extended to boolean formulas in the usual way.*



Dually, for an application condition ac_L *over* L *we define* $\mathrm{R}(p, \mathrm{ac}_L) = \mathrm{L}(p^{-1}, \mathrm{ac}_L)$*, where the* inverse rule p^{-1} *without application conditions is defined by* $p^{-1} = (R \overset{r}{\leftarrow} K \overset{l}{\rightarrow} L)$.

Fact 2. *Given a transformation* $G \overset{p,m}{\Longrightarrow} H$ *via a rule* $p = (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R, \mathrm{ac})$ *and an application condition* ac_R *over* R*, then* $m \models \mathrm{L}(p, \mathrm{ac}_R)$ *if and only if* $n \models \mathrm{ac}_R$.

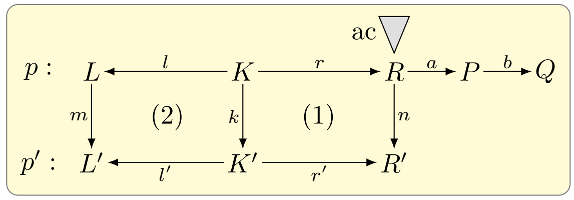


Dually, for an application condition ac_L *over* L *we have that* $m \models \mathrm{ac}_L$ *if and only if* $n \models \mathrm{R}(p, \mathrm{ac}_L)$.

Proof. See [13]. □

Shifts over morphisms are compositional and shifts over morphisms and rules are compatible via double pushouts.

Fact 3. *Given an application condition* ac *on* R, *the double pushouts* (1) *and* (2) *and morphisms* a, b, *then we have that*



– $\mathrm{Shift}(b, \mathrm{Shift}(a, \mathrm{ac})) \cong \mathrm{Shift}(b \circ a, \mathrm{ac})$,
– $\mathrm{Shift}(m, \mathrm{L}(p, \mathrm{ac})) \cong \mathrm{L}(p', \mathrm{Shift}(n, \mathrm{ac}))$.

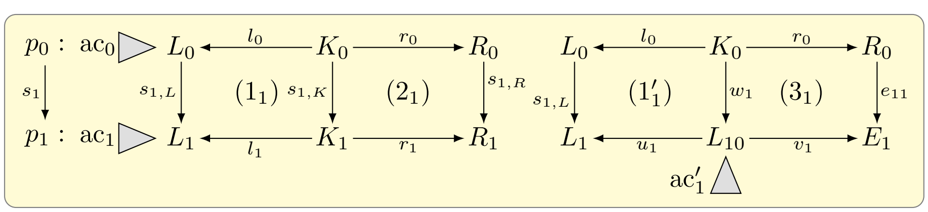
Proof. See [13, 14]. □

3 Decomposition of Direct Transformations

In this section, we show how to decompose a direct transformation in $\mathcal{M}$-adhesive categories into transformations via a kernel and a complement rule leading to the Complement Rule Theorem.

A kernel morphism describes how a smaller rule, the kernel rule, is embedded into a larger rule, the multi rule, which has its name because it can be applied multiple times for a given kernel rule match as described in Section 4. We need some more technical preconditions to make sure that the embeddings of the L-, K-, and R-components as well as the application conditions are consistent and allow to construct a complement rule.

Definition 7 (Kernel morphism). *Given rules* $p_0 = (L_0 \xleftarrow{l_0} K_0 \xrightarrow{r_0} R_0, \mathrm{ac}_0)$ *and* $p_1 = (L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1, \mathrm{ac}_1)$, *a* kernel morphism $s_1 : p_0 \to p_1$, $s_1 = (s_{1,L}, s_{1,K}, s_{1,R})$ *consists of* $\mathcal{M}$*-morphisms* $s_{1,L} : L_0 \to L_1$, $s_{1,K} : K_0 \to K_1$, *and* $s_{1,R} : R_0 \to R_1$ *such that in the following diagram* (1_1) *and* (2_1) *are pullbacks,* (1_1) *has a pushout complement* $(1'_1)$ *for* $s_{1,L} \circ l_0$, *and* ac_0 *and* ac_1 *are* complement-compatible w.r.t. s_1, *i.e. given pushout* (3_1) *then* $\mathrm{ac}_1 \cong \mathrm{Shift}(s_{1,L}, \mathrm{ac}_0) \wedge \mathrm{L}(p^*_1, \mathrm{Shift}(v_1, \mathrm{ac}'_1))$ *for some* ac'_1 *on* L_{10} *and* $p^*_1 = (L_1 \xleftarrow{u_1} L_{10} \xrightarrow{v_1} E_1)$. *In this case,* p_0 *is called* kernel rule *and* p_1 multi rule.



Remark 1. The complement-compatibility of the application conditions makes sure that there is a decomposition of ac_1 into parts on L_0 and L_{10}, where the latter ones are used later for the application conditions of the complement rule.

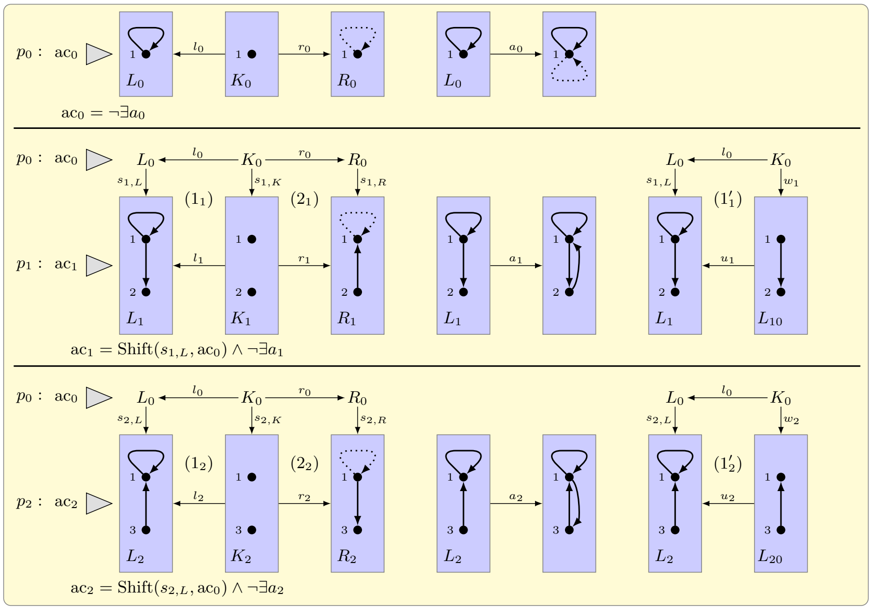


Fig. 1. The kernel rule p_0 and the multi rules p_1 and p_2

Example 1. To explain the concept of amalgamation, in our example we model a small transformation system for switching the direction of edges in labeled graphs, where we only have different labels for edges – black and dotted edges. The kernel rule p_0 is depicted in the top of Fig. 1. It selects a node with a black loop, deletes this loop, and adds a dotted loop, all of this if no dotted loop is already present. The matches are defined by the numbers at the nodes and can be induced for the edges by their position.

In the middle and bottom of Figure 1, two multi rules p_1 and p_2 are shown, which extend the rule p_0 and in addition reverse an edge if no backward edge is present. They also inherit the application condition of p_0 forbidding a dotted loop at the selected node. There is a kernel morphism $s_1 : p_0 \to p_1$ as shown in the top of Fig. 1 with pullbacks (1_1) and (2_1), and pushout complement $(1'_1)$. For the application conditions, $\mathrm{ac}_1 = \mathrm{Shift}(s_{1,L}, \mathrm{ac}_0) \wedge \neg\exists a_1 \cong \mathrm{Shift}(s_{1,L}, \mathrm{ac}_0) \wedge \mathrm{L}(p_1^*, \mathrm{Shift}(v_1, \neg\exists a'_1))$ with a'_1 as shown in the left of Fig. 2. We have that $\mathrm{Shift}(v_1, \neg\exists a'_1) = \neg\exists a_{11}$, because square $(*)$ is the only possible commuting square leading to a_{11}, b_{11} jointly surjective and b_{11} injective. Moreover, $\mathrm{L}(p_1^*, \neg\exists a_{11}) = \neg\exists a_1$ as shown by the two pushout squares (PO_1) and (PO_2) in Fig. 2. Thus $\mathrm{ac}'_1 = \neg\exists a'_1$, and ac_0 and ac_1 are complement compatible.

Similarly, there is a kernel morphism $s_2 : p_0 \to p_2$ as shown in the bottom of Fig. 1 with pullbacks (1_2) and (2_2), pushout complement $(1'_2)$, and ac_0 and ac_2 are complement compatible.

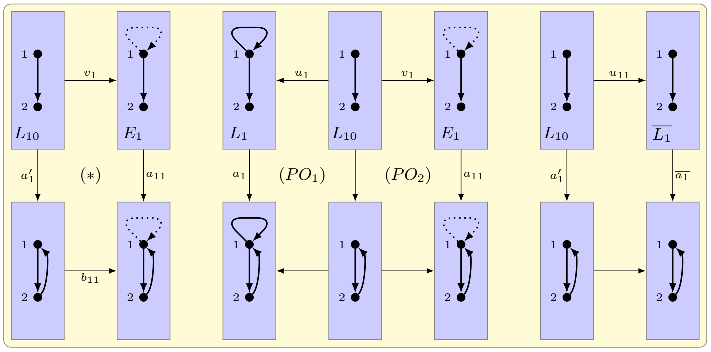
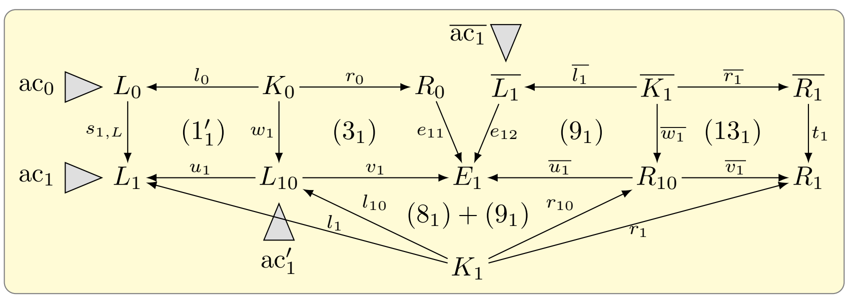


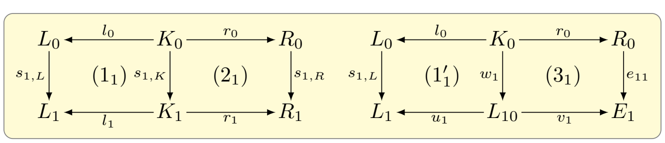
Fig. 2. Constructions for the application conditions

For a given kernel morphism, the complement rule is the remainder of the multi rule after the application of the kernel rule, i.e. it describes what the multi rule does in addition to the kernel rule.

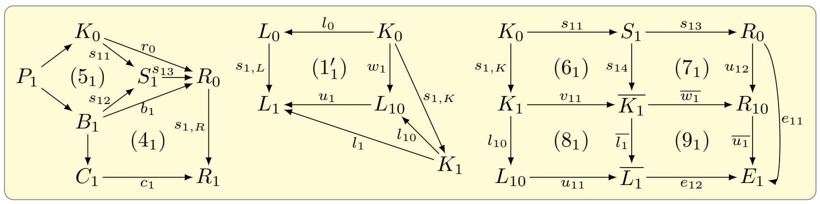
Theorem 1 (Existence of complement rule). *Given rules* $p_0 = (L_0 \xleftarrow{l_0} K_0 \xrightarrow{r_0} R_0, \mathrm{ac}_0)$ *and* $p_1 = (L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1, \mathrm{ac}_1)$*, and a kernel morphism* $s_1 : p_0 \to p_1$ *then there exists a rule* $\overline{p_1} = (\overline{L_1} \xleftarrow{\overline{l_1}} \overline{K_1} \xrightarrow{\overline{r_1}} \overline{R_1}, \overline{\mathrm{ac}_1})$ *and a jointly epimorphic cospan* $R_0 \xrightarrow{e_{11}} E_1 \xleftarrow{e_{12}} \overline{L_1}$ *such that the* E_1*-concurrent rule* $p_0 *_{E_1} \overline{p_1}$ *exists and* $p_1 = p_0 *_{E_1} \overline{p_1}$*. (For the definition of* E*-concurrent rules for rules with application conditions see [14].)*



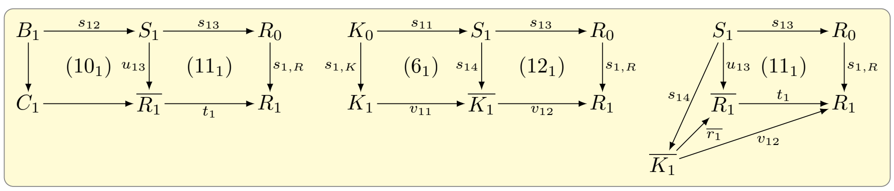
Proof. First, we consider the construction without application conditions. Since s_1 is a kernel morphism the following diagrams (1_1) and (2_1) are pullbacks, and (1_1) has a pushout complement $(1'_1)$ for $s_{1,L} \circ l_0$. Construct the pushout (3_1).



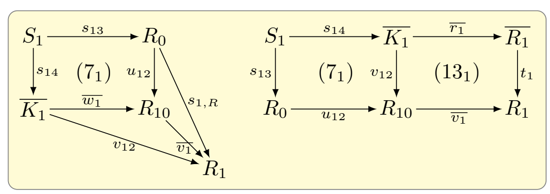
Now construct the initial pushout (4_1) over $s_{1,R}$ with $b_1, c_1 \in \mathcal{M}$, P_1 as the pullback object of r_0 and b_1, and the pushout (5_1) where we obtain an induced morphism $s_{13} : S_1 \to R_0$ with $s_{13} \circ s_{12} = b_1$, $s_{13} \circ s_{11} = r_0$, and $s_{13} \in \mathcal{M}$ by effective pushouts. Since (1_1) is a pullback Lemma A.1 implies that there is a unique morphism $l_{10} : K_1 \to L_{10}$ with $l_{10} \circ s_{1,K} = w_1$, $u_1 \circ l_{10} = l_1$, and $l_{10} \in \mathcal{M}$, and we can construct pushouts $(6_1) - (9_1)$ as a decomposition of pushout (3_1) which leads to $\overline{L_1}$ and $\overline{K_1}$ of the complement rule, and with $(7_1) + (9_1)$ being a pushout e_{11} and e_{12} are jointly epimorphic.



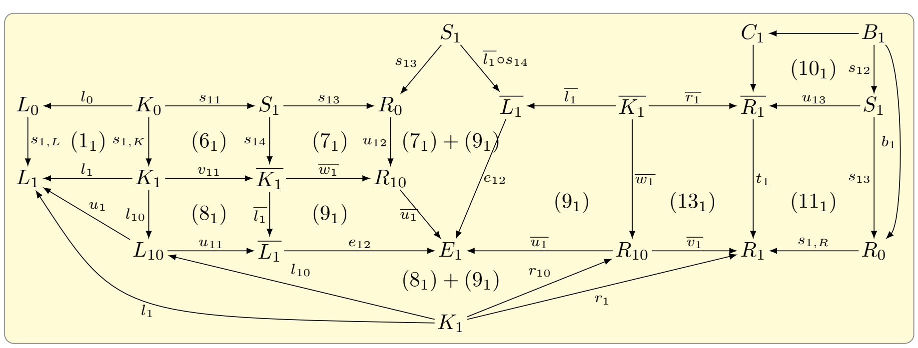
The pushout (4_1) can be decomposed into pushouts (10_1) and (11_1) obtaining the right hand side $\overline{R_1}$ of the complement rule, while pullback (2_1) can be decomposed into pushout (6_1) and square (12_1) which is a pullback by Lemma A.2. Now Lemma A.1 implies that there is a unique morphism $\overline{r_1} : \overline{K_1} \to \overline{R_1}$ with $\overline{r_1} \circ s_{14} = u_{13}$, $t_1 \circ \overline{r_1} = v_{12}$, and $\overline{r_1} \in \mathcal{M}$.



Now the pushout (7_1) implies that there is a unique morphism $\overline{v_1} : R_{10} \to R_1$, and by pushout decomposition of $(11_1) = (7_1) + (13_1)$ square (13_1) is a pushout.



Moreover, $(8_1) + (9_1)$ as a pushout over $\mathcal{M}$-morphisms is also a pullback which completes the construction, leading to the required rule $\overline{p_1} = (\overline{L_1} \xleftarrow{\overline{l_1}} \overline{K_1} \xrightarrow{\overline{r_1}} \overline{R_1})$ and $p_1 = p_0 *_{E_1} \overline{p_1}$ for rules without application conditions.



For the application conditions, suppose $\text{ac}_1 \cong \text{Shift}(s_{1,L}, \text{ac}_0) \wedge \text{L}(p_1^*, \text{Shift}(v_1, \text{ac}_1'))$ for $p_1^* = (L_1 \xleftarrow{u_1} L_{10} \xrightarrow{v_1} E_1)$ with $v_1 = e_{12} \circ u_{11}$ and ac_1' on L_{10}. Now define $\overline{\text{ac}_1} = \text{Shift}(u_{11}, \text{ac}_1')$, which is an application condition on $\overline{L_1}$.

We have to show that $(p_1, \text{ac}_{p_0 *_{E_1} \overline{p_1}}) \cong (p_1, \text{ac}_1)$. By construction of the E_1-concurrent rule we have that $\text{ac}_{p_0 *_{E_1} \overline{p_1}} \cong \text{Shift}(s_{1,L}, \text{ac}_0) \wedge \text{L}(p_1^*, \text{Shift}(e_{12}, \overline{\text{ac}_1})) \cong \text{Shift}(s_{1,L}, \text{ac}_0) \wedge \text{L}(p_1^*, \text{Shift}(e_{12}, \text{Shift}(u_{11}, \text{ac}_1'))) \cong \text{Shift}(s_{1,L}, \text{ac}_0) \wedge \text{L}(p_1^*, \text{Shift}(e_{12} \circ u_{11}, \text{ac}_1')) \cong \text{Shift}(s_{1,L}, \text{ac}_0) \wedge \text{L}(p_1^*, \text{Shift}(v_1, \text{ac}_1')) \cong \text{ac}_1$. □

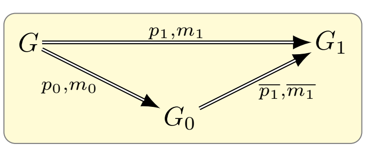
Remark 2. Note, that by construction the interface K_0 of the kernel rule has to be preserved in the complement rule. The construction of $\overline{p_1}$ is not unique w.r.t. the property $p_1 = p_0 *_{E_1} \overline{p_1}$, since other choices for S_1 with $\mathcal{M}$-morphisms s_{11} and s_{13} also lead to a well-defined construction. In particular, one could choose $S_1 = R_0$ leading to $\overline{p_1} = E_1 \xleftarrow{\overline{u_1}} R_{10} \xrightarrow{\overline{v_1}} R_1$. Our choice represents a smallest possible complement, which should be preferred in most application areas.

Definition 8 (Complement rule). *Given rules* $p_0 = (L_0 \xleftarrow{l_0} K_0 \xrightarrow{r_0} R_0, \text{ac}_0)$ *and* $p_1 = (L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1, \text{ac}_1)$*, and a kernel morphism* $s_1 : p_0 \to p_1$ *then the rule* $\overline{p_1} = (\overline{L_1} \xleftarrow{\overline{l_1}} \overline{K_1} \xrightarrow{\overline{r_1}} \overline{R_1}, \overline{\text{ac}_1})$ *constructed in Thm. 1 is called* complement rule *(of* s_1*).*

Example 2. Consider the kernel morphism s_1 depicted in Fig. 1. Using the construction in Thm. 1 we obtain the diagrams in Fig. 3 leading to the complement rule in the top row in Fig. 4 with the application condition $\overline{\text{ac}_1} = \neg\exists\overline{a_1}$ constructed in the right of Fig. 2. Similarly, we obtain a complement rule for the kernel morphism $s_2 : p_0 \to p_2$ in Fig. 1, which is depicted in the bottom row of Fig. 4.

Each direct transformation via a multi rule can be decomposed into a direct transformation via the kernel rule followed by a direct transformation via the complement rule.

Fact 4 *Given rules* $p_0 = (L_0 \xleftarrow{l_0} K_0 \xrightarrow{r_0} R_0, \text{ac}_0)$ *and* $p_1 = (L_1 \xleftarrow{l_1} K_1 \xrightarrow{r_1} R_1, \text{ac}_1)$*, a kernel morphism* $s_1 : p_0 \to p_1$*, and a direct transformation* $t_1 : G \xRightarrow{p_1, m_1} G_1$ *then* t_1 *can be decomposed into the transformation* $G \xRightarrow{p_0, m_0} G_0 \xRightarrow{\overline{p_1}, \overline{m_1}} G_1$ *with* $m_0 = m_1 \circ s_{1,L}$ *where* $\overline{p_1}$ *is the complement rule of* s_1*.*



Proof. We have that $p_1 \cong p_0 *_{E_1} \overline{p_1}$. The analysis part of the Concurrency Theorem [14] now implies the decomposition into $G \xRightarrow{p_0, m_0} G_0 \xRightarrow{\overline{p_1}, \overline{m_1}} G_1$ with $m_0 = m_1 \circ s_{1,L}$. □

4 Multi-Amalgamation

In [4], an Amalgamation Theorem for a pair of graph rules without application conditions has been developed. It can be seen as a generalization of the Parallelism Theorem [5], where the assumption of parallel independence is dropped

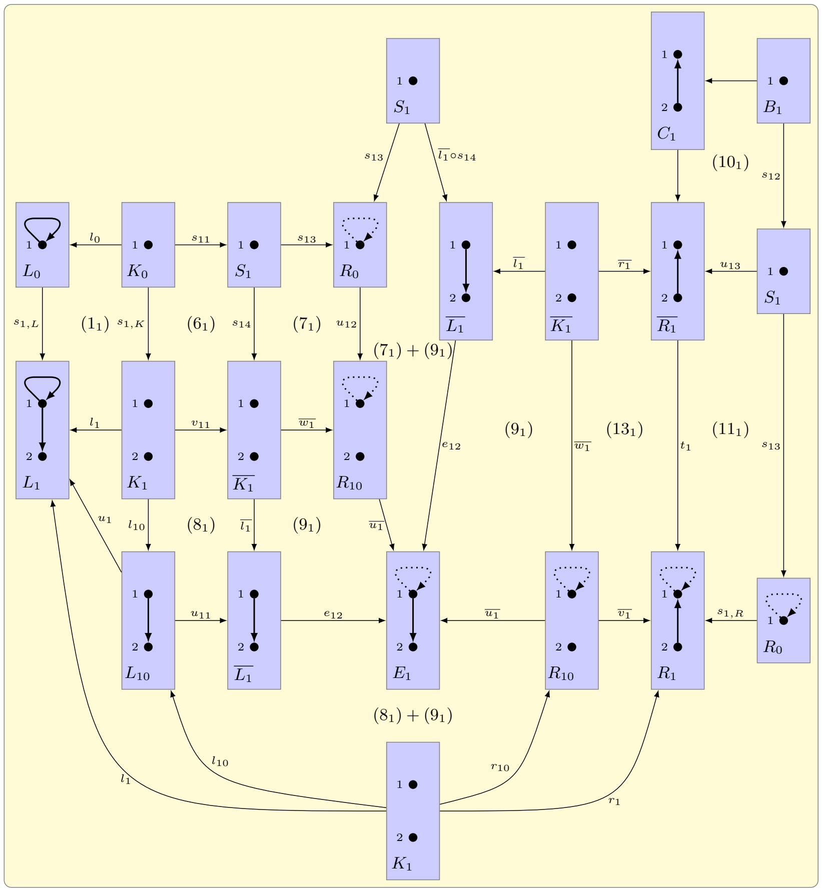


Fig. 3. The construction of the complement rule for the kernel morphism s_1

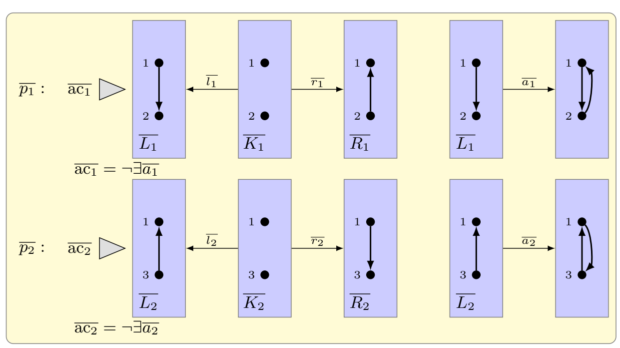


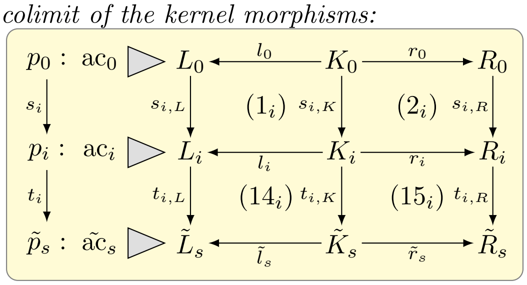
Fig. 4. The complement rules for the kernel morphisms

and pure parallelism is generalized to synchronized parallelism. In this section, we present an Amalgamation Theorem for a bundle of rules with application conditions, called Multi-Amalgamation Theorem, over objects in an $\mathcal{M}$-adhesive category.

We consider not only single kernel morphisms, but bundles of them over a fixed kernel rule. Then we can combine the multi rules of such a bundle to an amalgamated rule by gluing them along the common kernel rule.

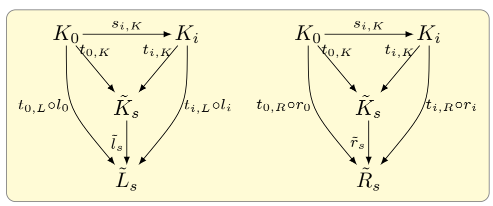
Definition 9 (Multi-amalgamated rule). *Given rules* $p_i = (L_i \xleftarrow{l_i} K_i \xrightarrow{r_i} R_i, \mathrm{ac}_i)$ *for* $i = 0, \ldots, n$ *and a bundle of kernel morphisms* $s = (s_i : p_0 \rightarrow p_i)_{i=1,\ldots,n}$*, then the* (multi-)amalgamated rule $\tilde{p}_s = (\tilde{L}_s \xleftarrow{\tilde{l}_s} \tilde{K}_s \xrightarrow{\tilde{r}_s} \tilde{R}_s, \tilde{\mathrm{ac}}_s)$ *is constructed as the componentwise colimit of the kernel morphisms:*

- $\tilde{L}_s = Col((s_{i,L})_{i=1,\ldots,n})$,
- $\tilde{K}_s = Col((s_{i,K})_{i=1,\ldots,n})$,
- $\tilde{R}_s = Col((s_{i,R})_{i=1,\ldots,n})$,
- $\tilde{l}_s$ *and* $\tilde{r}_s$ *are induced by* $(t_{i,L} \circ l_i)_{i=0,\ldots,n}$ *and* $(t_{i,R} \circ r_i)_{0=1,\ldots,n}$*, respectively,*
- $\tilde{\mathrm{ac}}_s = \bigwedge_{i=1,\ldots,n} \mathrm{Shift}(t_{i,L}, \mathrm{ac}_i)$.



Fact 5 *The amalgamated rule is well-defined and we have kernel morphisms* $t_i = (t_{i,L}, t_{i,K}, t_{i,R}) : p_i \rightarrow \tilde{p}_s$ *for* $i = 0, \ldots, n$.

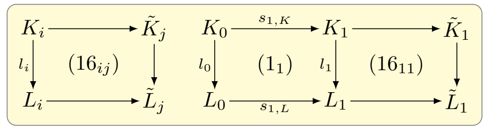
Proof.
First we show the well-definedness of the morphism $\tilde{l}_s$. Consider the colimits $(\tilde{L}_s, (t_{i,L})_{i=0,\ldots,n})$ of $(s_{i,L})_{i=1,\ldots,n}$, $(\tilde{K}_s, (t_{i,K})_{i=0,\ldots,n})$ of $(s_{i,K})_{i=1,\ldots,n}$, and $(\tilde{R}_s, (t_{i,R})_{i=0,\ldots,n})$ of $(s_{i,R})_{i=1,\ldots,n}$, with $t_{0,*} = t_{i,*} \circ s_{i,*}$ for $* \in \{L, K, R\}$. Since $t_{i,L} \circ l_i \circ s_{i,K} = t_{i,L} \circ s_{i,L} \circ l_0 = t_{0,L} \circ l_0$, we get an induced morphism $\tilde{l}_s : \tilde{K}_s \rightarrow \tilde{L}_s$ with $\tilde{l}_s \circ t_{i,K} = t_{i,L} \circ l_i$ for $i = 0, \ldots, n$. Similarly, we obtain $\tilde{r}_s : \tilde{K}_s \rightarrow \tilde{R}_s$ with $\tilde{r}_s \circ t_{i,K} = t_{i,R} \circ r_i$ for $i = 0, \ldots, n$.



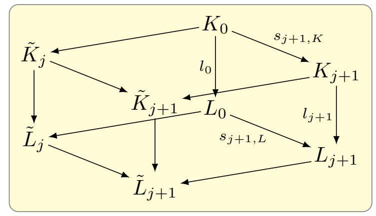
The colimit of a bundle of n morphisms can be constructed by iterated pushout constructions, which means that we only have to require pushouts over $\mathcal{M}$-morphisms. Since pushouts are closed under $\mathcal{M}$-morphisms, the iterated pushout construction leads to $t \in \mathcal{M}$.

It remains to show that (14_i) resp. $(14_i) + (1_i)$ and (15_i) resp. $(15_i) + (2_i)$ are pullbacks, and (14_i) resp. $(14_i) + (1_i)$ has a pushout complement for $t_{i,L} \circ l_i$. We prove this by induction over j for (14_i) resp. $(14_i) + (1_i)$, the pullback property of (15_i) follows analogously.

We prove: Let $\tilde{L}_j$ and $\tilde{K}_j$ be the colimits of $(s_{i,L})_{i=1,\ldots,j}$ and $(s_{i,K})_{i=1,\ldots,j}$, respectively. Then (16_{ij}) is a pullback with pushout complement property for all $i = 0, \ldots, j$.

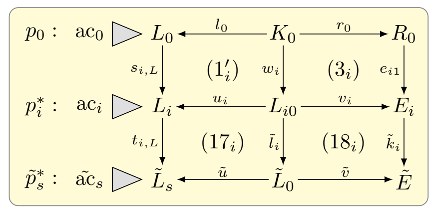


Basis $j = 1$: The colimits of $s_{1,L}$ and $s_{1,K}$ are L_1 and K_1, respectively, which means that $(16_{01}) = (1) + (16_{11})$ and (16_{11}) are both pushouts and pullbacks.



Induction step $j \to j+1$: Construct $\tilde{L}_{j+1} = \tilde{L}_j +_{L_0} L_{j+1}$ and $\tilde{K}_{j+1} = \tilde{K}_j +_{K_0} K_{j+1}$ as pushouts, and we have the following cube with the top and bottom faces as pushouts, the back faces as pullbacks, and by the van Kampen property also the front faces are pullbacks. Moreover, by Lemma A.3 the front faces have the pushout complement property, and by Lemma A.4 this holds also for (16_{0j}) and (16_{ij}) as compositions.

Thus, for a given n, (16_{in}) is the required pullback (14_i) resp. $(14_i) + (1_i)$ with pushout complement property, using $\tilde{K}_n = \tilde{K}_s$ and $\tilde{L}_n = \tilde{L}_s$.



Moreover, we have pushout complements (17_i) resp. $(17_i) + (1'_i)$ for $t_{i,L} \circ l_i$. Since ac_0 and ac_i are complement-compatible for all i we have that $\mathrm{ac}_i \cong \mathrm{Shift}(s_{i,L}, \mathrm{ac}_0) \wedge \mathrm{L}(p_i^*, \mathrm{Shift}(v_i, \mathrm{ac}'_i))$. For any ac'_i, it holds that $\mathrm{Shift}(t_{i,L}, \mathrm{L}(p_i^*, \mathrm{Shift}(v_i, \mathrm{ac}'_i)))) \cong \mathrm{L}(\tilde{p}_s^*, \mathrm{Shift}(\tilde{k}_i \circ v_i, \mathrm{ac}'_i)) \cong \mathrm{L}(\tilde{p}_s^*, \mathrm{Shift}(\tilde{v}, \mathrm{Shift}(\tilde{l}_i, \mathrm{ac}'_i)))$, since all squares are pushouts by pushout-pullback decomposition and the uniqueness of pushout complements. Define $\mathrm{ac}_i^* := \mathrm{Shift}(\tilde{l}_i, \mathrm{ac}'_i)$ as an application condition on $\tilde{L}_0$. It follows that $\tilde{\mathrm{ac}}_s = \bigwedge_{i=1,\ldots,n} \mathrm{Shift}(t_{i,L}, \mathrm{ac}_i) \cong \bigwedge_{i=1,\ldots,n} (\mathrm{Shift}(t_{i,L} \circ s_{i,L}, \mathrm{ac}_0) \wedge \mathrm{Shift}(t_{i,L}, \mathrm{L}(p_i^*, \mathrm{Shift}(v_i, \mathrm{ac}'_i)))) \cong \mathrm{Shift}(t_{0,L}, \mathrm{ac}_0) \wedge \bigwedge_{i=1,\ldots,n} \mathrm{L}(\tilde{p}_s^*, \mathrm{Shift}(\tilde{v}, \mathrm{ac}_i^*))$.

For $i = 0$ define $\mathrm{ac}'_{s0} = \bigwedge_{j=1,\ldots,n} ac_j^*$, and hence $\tilde{\mathrm{ac}}_s = \mathrm{Shift}(t_{0,L}, \mathrm{ac}_0) \wedge \mathrm{L}(\tilde{p}_s^*, \mathrm{Shift}(\tilde{v}, \mathrm{ac}'_{s0}))$ implies the complement-compatibility of ac_0 and $\tilde{\mathrm{ac}}_s$. For $i > 0$, we have that $\mathrm{Shift}(t_{0,L}, \mathrm{ac}_0) \wedge \mathrm{L}(\tilde{p}_s^*, \mathrm{Shift}(\tilde{v}, \mathrm{ac}_i^*)) \cong \mathrm{Shift}(t_{i,L}, \mathrm{ac}_i)$. Define $\mathrm{ac}'_{si} = \bigwedge_{j=1,\ldots,n \setminus i} ac_j^*$, and hence $\tilde{\mathrm{ac}}_s = \mathrm{Shift}(t_{i,L}, \mathrm{ac}_i) \wedge \mathrm{L}(\tilde{p}_s^*, \mathrm{Shift}(\tilde{v}, \mathrm{ac}'_{si}))$ implies the complement-compatibility of ac_i and $\tilde{\mathrm{ac}}_s$. □

The application of an amalgamated rule yields an amalgamated transformation.

Definition 10 (Amalgamated transformation). *The application of an amalgamated rule to a graph G is called an* amalgamated transformation*.*

Example 3. Consider the bundle $s = (s_1, s_2, s_3 = s_1)$ of the kernel morphisms depicted in Fig. 1. The corresponding amalgamated rule $\tilde{p}_s$ is shown in the top row of Fig. 5. This amalgamated rule can be applied to the graph G leading to the amalgamated transformation depicted in Fig. 5, where the application condition $\tilde{ac}_s$ is obviously fulfilled by the match $\tilde{m}$.

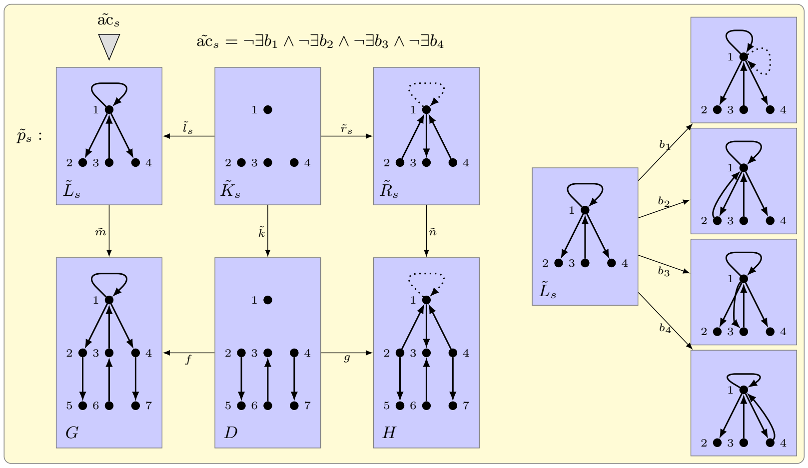
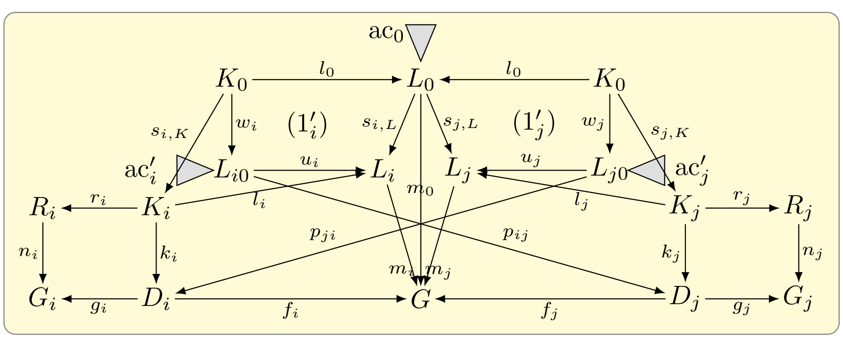


Fig. 5. An amalgamated transformation

If we have a bundle of direct transformations of a graph G, where for each transformation one of the multi rules is applied, we want to analyze if the amalgamated rule is applicable to G combining all the single transformation steps. These transformations are compatible, i.e. multi-amalgamable, if the matches agree on the kernel rules, and are independent outside.

Definition 11 (Multi-amalgamable). *Given a bundle of kernel morphisms $s = (s_i : p_0 \to p_i)_{i=1,\dots,n}$, a bundle of direct transformations steps $(G \xRightarrow{p_i,m_i} G_i)_{i=1,\dots n}$ is* s-multi-amalgamable*, or short s-amalgamable, if*

- *it has* consistent matches*, i.e. $m_i \circ s_{i,L} = m_j \circ s_{j,L} =: m_0$ for all $i, j = 1, \dots, n$ and*
- *it has* weakly independent matches*, i.e. for all $i \neq j$ consider the pushout complements $(1'_i)$ and $(1'_j)$, and then there exist morphisms $p_{ij} : L_{i0} \to D_j$ and $p_{ji} : L_{j0} \to D_i$ such that $f_j \circ p_{ij} = m_i \circ u_i$, $f_i \circ p_{ji} = m_j \circ u_j$, $g_j \circ p_{ij} \models \mathrm{ac}'_i$, and $g_i \circ p_{ji} \models \mathrm{ac}'_j$.*



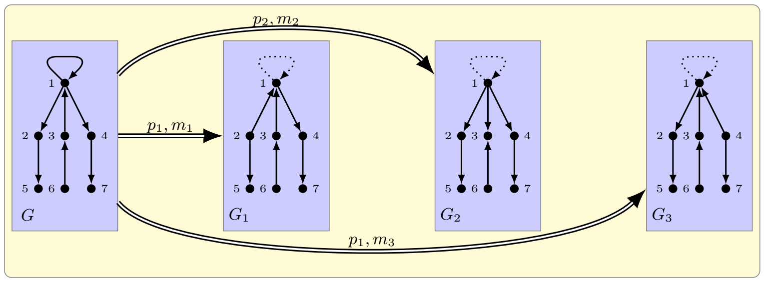
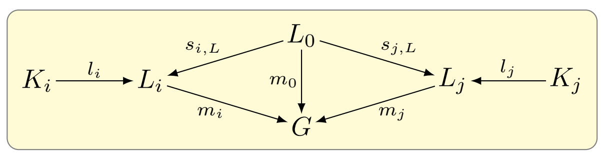


Fig. 6. An s-amalgamable bundle of direct transformations

Similar to the characterization of parallel independence in [2] we can give a set-theoretical characterization of weak independence.

Fact 6 *For graphs and other set-based structures, weakly independent matches means that* $m_i(L_i) \cap m_j(L_j) \subseteq m_0(L_0) \cup (m_i(l_i(K_i)) \cap m_j(l_j(K_j)))$ *for all* $i \neq j = 1, \ldots, n$*, i.e. the elements in the intersection of the matches* m_i *and* m_j *are either preserved by both transformations, or are also matched by* m_0*.*



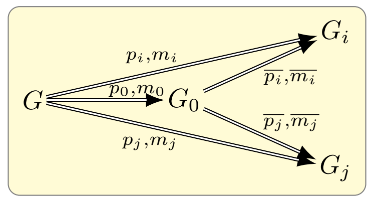
Proof. We have to proof the equivalence of $m_i(L_i) \cap m_j(L_j) \subseteq m_0(L_0) \cup (m_i(l_i(K_i)) \cap m_j(l_j(K_j)))$ for all $i \neq j = 1, \ldots, n$ with the definition of weakly independent matches.

"$\Leftarrow$" Let $x = m_i(y_i) = m_j(y_j)$, and suppose $x \notin m_0(L_0)$. Since $(1'_i)$ is a pushout we have that $y_i = u_i(z_i) \in u_i(L_{i0} \backslash w_i(K_0))$, and $x = m_i(u_i(z_i)) = f_j(p_i(z_i)) = m_j(y_j)$, and by pushout properties $y_j \in l_j(K_j)$ and $x \in m_j(l_j(K_j))$. Similarly, $x \in m_i(l_i(K_i))$.

"$\Rightarrow$" For $x \in L_{i0}$, $x = w_i(k)$ define $p_{ij}(x) = k_j(s_{j,K}(k))$, then $f_j(p_{ij}(x)) = f_j(k_j(s_{j,K}(k))) = m_j(l_j(s_{j,K}(k))) = m_j(s_{j,L}(l_(k))) = m_i(s_{i,L}(l_0(k))) = m_i(n_i(w_i(k))) = m_i(u_i(x))$. Otherwise, $x \notin w_i(K_0)$, i.e. $u_i(x) \notin s_{i,L}(L_0)$, and define $p_{ij}(x) = y$ with $f_j(y) = m_i(u_i(x))$. This y exists, because either $m_i(u_i(x)) \notin m_j(L_j)$ or $m_i(u_i(x)) \in m_j(L_j)$ and then $m_i(u_i(x)) \in m_j(l_j(K_j))$, and in both cases $m_i(u_i(x)) \in f_j(D_j)$. Similarly, we can define p_{ji} with the required property. □

Example 4. Consider the bundle $s = (s_1, s_2, s_3 = s_1)$ of kernel morphisms from Ex. 3. For the graph G given in Fig. 5 we find matches $m_0 : L_0 \to G$, $m_1 : L_1 \to G$, $m_2 : L_2 \to G$, and $m_3 : L_1 \to G$ mapping all nodes from the left hand side to their corresponding nodes in G, except for m_3 mapping node 2 in L_1 to node 4 in G. For all these matches, the corresponding application conditions are fulfilled and we can apply the rules p_1, p_2, p_1, respectively, leading to the bundle of direct transformations depicted in Fig. 6. This bundle is s-amalgamable, because the matches m_1, m_2, and m_3 agree on the match m_0, and are weakly independent, because they only overlap in m_0.

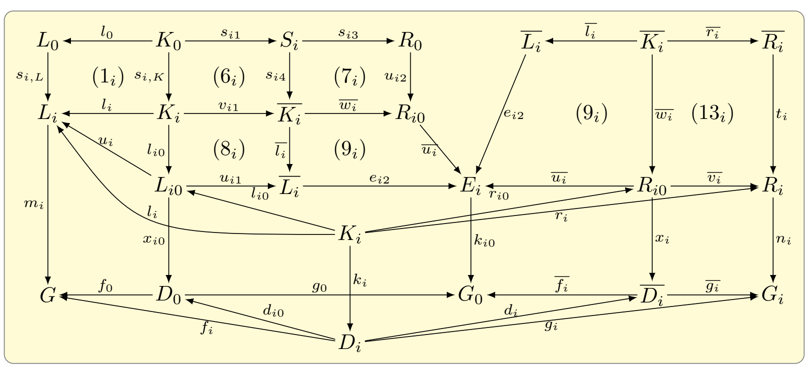
For an s-amalgamable bundle of direct transformations, each single transformation step can be decomposed into an application of the kernel rule followed by an application of the complement rule. Moreover, all kernel rule applications lead to the same object, and the following applications of the complement rules are parallel independent.



Fact 7. *Given a bundle of kernel morphisms $s = (s_i : p_0 \to p_i)_{i=1,\dots,n}$ and an s-amalgamable bundle of direct transformations $(G \xRightarrow{p_i,m_i} G_i)_{i=1,\dots,n}$ then each direct transformation $G \xRightarrow{p_i,m_i} G_i$ can be decomposed into a transformation $G \xRightarrow{p_0,m_0} G_0 \xRightarrow{\overline{p_i},\overline{m_i}} G_i$. Moreover, the transformations $G_0 \xRightarrow{\overline{p_i},\overline{m_i}} G_i$ are pairwise parallel independent.*

Proof. From Fact 4 it follows that each single direct transformation $G \xRightarrow{p_i,m_i} G_i$ can be decomposed into a transformation $G \xRightarrow{p_0,m_0^i} G_0^i \xRightarrow{\overline{p_i},\overline{m_i}} G_i$ with $m_0^i = m_i \circ s_{i,L}$, and since the bundle is s-amalgamable, $m_0 = m_i \circ s_{i,L} = m_0^i$ and $G_0 := G_0^i$ for all $i = 1, \dots, n$.

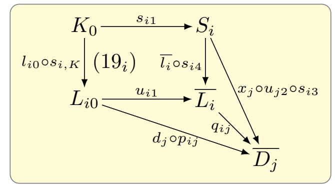
We have to show the pairwise parallel independence. From the constructions of the complement rule and the Concurrency Theorem we obtain the following diagram for all $i = 1, \dots, n$.



For $i \neq j$, from weakly independent matches it follows that we have a morphism $p_{ij} : L_{i0} \to D_j$ with $f_j \circ p_{ij} = m_i \circ u_i$. It follows that $f_j \circ p_{ij} \circ w_i = m_i \circ u_i \circ w_i = m_i \circ s_{i,L} \circ l_0 = m_0 \circ l_0 = m_j \circ s_{j,L} \circ l_0 = m_j \circ u_j \circ w_j = m_j \circ u_j \circ l_{j0} \circ s_{j,K} = m_j \circ l_j \circ s_{j,K} = f_j \circ k_j \circ s_{j,K}$, and with $f_j \in \mathcal{M}$ we have that $p_{ij} \circ w_i = k_j \circ s_{jk}$ $(*)$.

Now consider the pushout $(19_i) = (6_i) + (8_i)$ in comparison with object $\overline{D_j}$ and morphisms $d_j \circ p_{ij}$ and $x_j \circ u_{j2} \circ s_{i3}$. We have that $d_j \circ p_{ij} \circ l_{i0} \circ s_{i,K} = d_j \circ p_{ij} \circ w_i \overset{(*)}{=} d_j \circ k_j \circ s_{j,K} = x_j \circ r_{j0} \circ s_{j,K} = x_j \circ \overline{w_j} \circ v_{j1} \circ s_{j,K} = x_j \circ u_{j2} \circ s_{j3} \circ s_{j1} =$

$x_j \circ u_{j2} \circ r_0 = x_j \circ u_{j2} \circ s_{i3} \circ s_{i1}$. Now pushout (18_i) induces a unique morphism q_{ij} with $q_{ij} \circ u_{i1} = d_j \circ p_{ij}$ and $q_{ij} \circ \overline{l_i} \circ s_{i4} = x_j \circ u_{j2} \circ s_{i3}$.



For the parallel independence of $G_0 \xRightarrow{\overline{p_i}, \overline{m_i}} \overline{G_i}$, $G_0 \xRightarrow{\overline{p_j}, \overline{m_j}} \overline{G_j}$, we have to show that $q_{ij} : \overline{L_i} \to \overline{D_j}$ satisfies $\overline{f_j} \circ q_{ij} = k_{i0} \circ e_{i2} =: \overline{m_i}$.

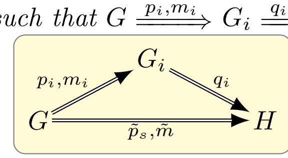
With $f_0 \in \mathcal{M}$ and $f_0 \circ d_{j0} \circ p_{ij} = f_j \circ p_{ij} = m_i \circ u_i = f_0 \circ c_{i0}$ it follows that $d_{j0} \circ p_{ij} = x_{i0}$ $(**)$. This means that $\overline{f_j} \circ q_{ij} \circ u_{i1} = \overline{f_j} \circ d_j \circ p_{ij} = \overline{g_0} \circ d_0 \circ p_{ij} \overset{(**)}{=} g_0 \circ x_{i0} = k_{i0} \circ e_{i2} \circ u_{i1}$. In addition, we have that $\overline{f_j} \circ q_{ij} \circ \overline{l_i} \circ s_{i4} = \overline{f_j} \circ x_j \circ u_{j2} \circ s_{i3} = k_{j0} \circ \overline{u_j} \circ u_{j2} \circ s_{i3} = k_{i0} \circ \overline{u_i} \circ u_{i2} \circ s_{i3} = k_{i0} \circ e_{i2} \circ \overline{l_i} \circ s_{i4}$. Since (19_i) is a pushout, u_{i1} and $\overline{l_i} \circ s_{i4}$ are jointly epimorphic, and it follows that $\overline{f_j} \circ q_{ij} \circ e_{i2} = k_{i0} \circ e_{i2}$.

If ac_0 and ac_i are not complement-compatible, then $\overline{\mathrm{ac}_i} = \mathrm{true}$ and trivially $\overline{g_j} \circ q_{ij} \models \overline{\mathrm{ac}_i}$ for all $j \neq i$. Otherwise, we have that $g_j \circ p_{ij} \models \mathrm{ac}'_i$, and with $g_j \circ p_{ij} = \overline{g_j} \circ d_j \circ p_{ij} = \overline{g_j} \circ q_{ij} \circ u_{i1}$ it follows that $\overline{g_j} \circ q_{ij} \circ u_{i1} \models \mathrm{ac}'_i$, which is equivalent to $\overline{g_j} \circ q_{ij} \models \mathrm{Shift}(u_{i1}, \mathrm{ac}'_1) = \overline{ac_i}$. □

If a bundle of direct transformations of a graph G is s-amalgamable, then we can apply the amalgamated rule directly to G leading to a parallel execution of all the changes done by the single transformation steps.

Theorem 2 (Multi-Amalgamation). *Consider a bundle of kernel morphisms $s = (s_i : p_0 \to p_i)_{i=1,\ldots,n}$.*

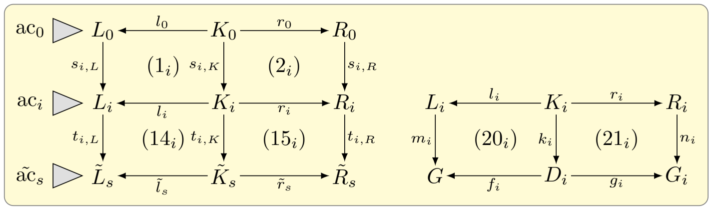
1. Synthesis. *Given an s-amalgamable bundle of direct transformations $(G \xRightarrow{p_i, m_i} G_i)_{i=1,\ldots,n}$ then there is an amalgamated transformation $G \xRightarrow{\tilde{p}_s, \tilde{m}} H$ and transformations $G_i \xRightarrow{q_i} H$ over the complement rules q_i of the kernel morphisms $t_i : p_i \to \tilde{p}_s$ such that $G \xRightarrow{p_i, m_i} G_i \xRightarrow{q_i} H$ is a decomposition of $G \xRightarrow{\tilde{p}_s, \tilde{m}} H$.*



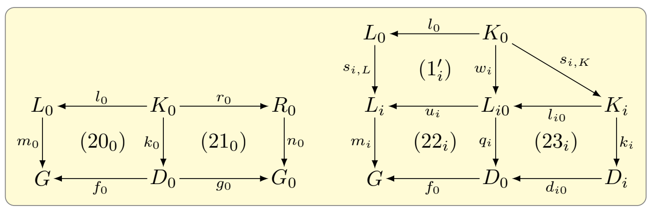
2. Analysis. *Given an amalgamated transformation $G \xRightarrow{\tilde{p}_s, \tilde{m}} H$ then there are s_i-related transformations $G \xRightarrow{p_i, m_i} G_i \xRightarrow{q_i} H$ for $i = 1, \ldots, n$ such that $G \xRightarrow{p_i, m_i} G_i$ is s-amalgamable.*
3. Bijective Correspondence. *The synthesis and analysis constructions are inverse to each other up to isomorphism.*

Proof. 1. Synthesis. We have to show that $\tilde{p}_s$ is applicable to G leading to an amalgamated transformation $G \xRightarrow{\tilde{p}_s, \tilde{m}} H$ with $m_i = \tilde{m} \circ t_{i,L}$, where $t_i : p_i \to \tilde{p}_i$ is the kernel morphism constructed in Fact 5. Then we can apply Fact 4 which implies the decomposition of $G \xRightarrow{\tilde{p}_s, \tilde{m}} H$ into $G \xRightarrow{p_i, m_i} G_i \xRightarrow{q_i} H$, where q_i is the (weak) complement rule of the kernel morphism t_i.

Given the kernel morphisms, the amalgamated rule, and the bundle of direct transformations, we have pullbacks (1_i), (2_i), (14_i), (15_i), and pushouts (20_i), (21_i).



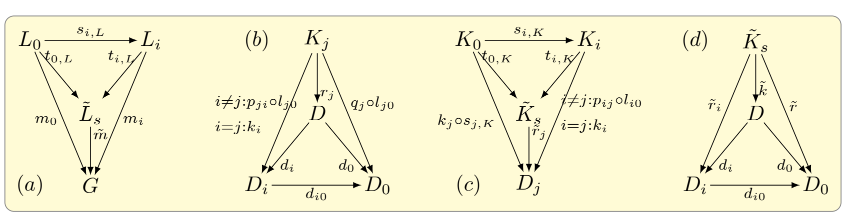
Using Fact 7, we know that we can apply p_0 via m_0 leading to a direct transformation $G \xRightarrow{p_0,m_0} G_0$ given by pushouts (20_0) and (21_0). Moreover, we find decompositions of pushouts (20_0) and (20_i) into pushouts $(1'_i)$ and (22_i) resp. (22_i) and (23_i) by $\mathcal{M}$-pushout pullback decomposition and uniqueness of pushout complements.



Since we have consistent matches, $m_i \circ s_{i,L} = m_0$ for all $i = 1, \ldots, n$. Then the colimit $\tilde{L}_s$ implies that there is a unique morphism $\tilde{m} : \tilde{L}_s \to G$ with $\tilde{m} \circ t_{i,L} = m_i$ and $\tilde{m} \circ t_{0,L} = m_0$ (a). Moreover, $m_i \models \mathrm{ac}_i \Rightarrow \tilde{m} \circ t_{i,L} \models \mathrm{ac}_i \Rightarrow \tilde{m} \models \mathrm{Shift}(t_{i,L}, \mathrm{ac}_i)$ for all $i = 1, \ldots, n$, and thus $\tilde{m} \models \tilde{\mathrm{ac}}_s = \bigwedge_{i=1,\ldots,n} \mathrm{Shift}(t_{i,L}, \mathrm{ac}_i)$

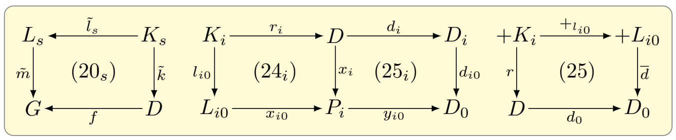
Weakly independent matches means that there exist morphisms p_{ij} with $f_j \circ p_{ij} = m_i \circ u_i$ for $i \neq j$. Construct D as the limit of $(d_{i0})_{i=1,\ldots,n}$ with morphisms d_i. Now f_0 being a monomorphism with $f_0 \circ d_{i0} \circ p_{ji} = f_i \circ p_{ji} = m_j \circ u_j = f_0 \circ q_j$ implies that $d_{i0} \circ p_{ji} = q_j$. It follows that $d_{i0} \circ p_{ji} \circ l_{j0} = q_j \circ l_{j0}$, and together with $d_{i0} \circ k_i = q_i \circ l_{i0}$ limit D implies that there exists a unique morphism r_j with $d_i \circ r_j = p_{ji} \circ l_{ji}$, $d_i \circ r_i = k_i$, and $d_0 \circ r_j = q_j \circ l_{j0}$ (b).

Similarly, f_j being a monomorphism with $f_j \circ p_{ij} \circ l_{i0} \circ s_{i,K} = m_i \circ u_i \circ w_i = m_i \circ s_{i,L} \circ l_0 = m_0 \circ l_0 = m_j \circ s_{j,L} \circ l_0 = m_j \circ l_j \circ s_{j,K} = f_j \circ k_j \circ s_{j,K}$ implies that $p_{ij} \circ l_{i0} \circ s_{i,K} = k_j \circ s_{j,K}$. Now colimit $\tilde{K}_s$ implies that there is a unique morphisms $\tilde{r}_j$ with $\tilde{r}_j \circ t_{i,K} = p_{ij} \circ l_{i0}$, $\tilde{r}_j \circ t_{j,K} = k_j$, and $\tilde{r}_j \circ t_{0,K} = k_j \circ s_{j,K}$ (c). Since $d_{i0} \circ \tilde{r}_i \circ t_{i,K} = d_{i0} \circ k_i = q_i \circ l_{i0} = d_{j0} \circ p_{ij} \circ l_{i0} = d_{j0} \circ \tilde{r}_j \circ t_{i,K}$ and $d_{i0} \circ \tilde{r}_i \circ t_{0,K} = d_{i0} \circ k_i \circ s_{i,K} = k_0 = d_{j0} \circ \tilde{r}_j \circ t_{0,K}$ colimit $\tilde{K}_s$ implies that for all i, j we have that $d_{i0} \circ \tilde{r}_i = d_{j0} \circ \tilde{r}_j =: \tilde{r}$. From limit D it now follows that there exists a unique morphism $\tilde{k}$ with $d_i \circ \tilde{k} = \tilde{r}_i$ and $d_0 \circ \tilde{k} = \tilde{r}$ (d).

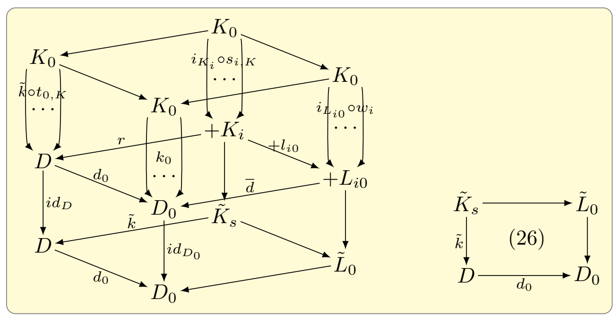


We have to show that (20_s) with $f = f_0 \circ d_0$ is a pushout. With $f \circ \tilde{k} \circ t_{i,K} = f_0 \circ d_0 \circ \tilde{k} \circ t_{i,K} = f_0 \circ \tilde{r} \circ t_{i,K} = f_0 \circ d_{i0} \circ \tilde{r}_i \circ t_{i,K} = f_0 \circ d_{i0} \circ k_i = f_i \circ k_i = m_i \circ l_i = \tilde{m} \circ t_{i,L} \circ l_i = \tilde{m} \circ \tilde{l}_s \circ t_{i,K}$ and $f \circ \tilde{k} \circ t_{0,K} = f_0 \circ d_0 \circ \tilde{k} \circ t_{0,K} = f_0 \tilde{r} \circ t_{0,K} = f_0 \circ d_{i0} \circ \tilde{r}_i \circ t_{0,K} = f_0 \circ d_{i0} \circ k_i \circ s_{i,K} = f_0 \circ k_0 = m_0 \circ l_0 = \tilde{m} \circ t_{0,L} \circ l_0 = \tilde{m} \circ \tilde{l}_s \circ t_{0,K}$ and $\tilde{K}_s$ being colimit it follows that $f \circ \tilde{k} = \tilde{m} \circ \tilde{l}_s$, thus the square commutes.

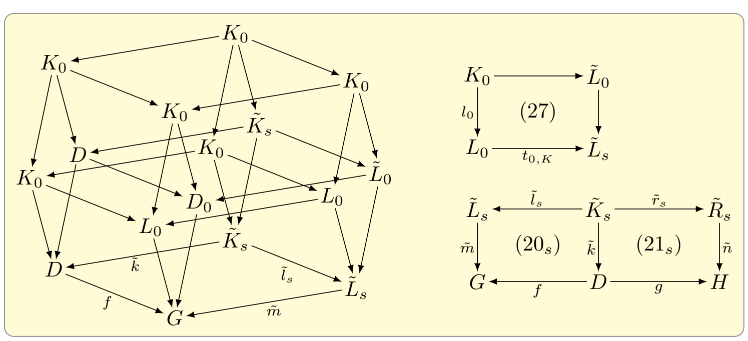
Pushout (23_i) can be decomposed into pushouts (24_i) and (25_i). Using Lemma A.5 it follows that D_0 is the colimit of $(x_i)_{i=1,\dots,n}$, because (23_i) is a pushout, D is the limit of $(d_{i0})_{i=1,\dots,n}$, and we have morphisms p_{ij} with $d_{j0} \circ p_{ij} = q_i$. Then Lemma A.6 implies that also (25) is a pushout.



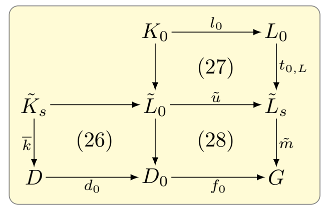
Now consider the coequalizers $\tilde{K}_s$ of $(i_{K_i} \circ s_{i,K} : K_0 \to +K_i)_{i=1,\dots,n}$ (which is actually $\tilde{K}_s$ by construction of colimits), $\tilde{L}_0$ of $(i_{L_{i0}} \circ w_i : K_0 \to +L_{i0})_{i=1,\dots,n}$ (as already constructed in Fact 5), D of $(\tilde{k} \circ t_{0,K} : K_0 \to D)_{i=1,\dots,n}$, and D_0 of $(k_0 : K_0 \to D_0)_{i=1,\dots,n}$. Consider the following cube, where the top square with identical morphisms is a pushout, the top cube commutes, and the middle square is pushout (25). Using Lemma A.7 it follows that also the bottom (26) constructed of the four coequalizers is a pushout.



Now consider the following cube, where the top and middle squares are pushouts and the two top cubes commute. Using again Lemma A.7 it follows that (20_s) in the bottom is actually a pushout, where $(27) = (1'_i) + (17_i)$ is a pushout by composition. Now we can construct pushout (21_s) which completes the direct transformation $G \overset{\tilde{p}_s, \tilde{m}}{\Longrightarrow} H$.

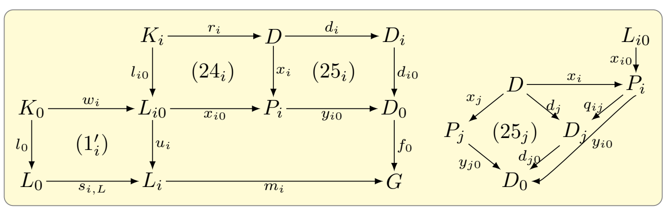


2. Analysis. Using the kernel morphisms t_i we obtain transformations $G \xRightarrow{p_i,m_i} G_i \xRightarrow{q_i} H$ from Fact 4 with $m_i = \tilde{m} \circ t_{i,L}$. We have to show that this bundle of transformation is s-amalgamable. Applying again Fact 4 we obtain transformations $G \xRightarrow{p_0,m_0^i} G_0^i \xRightarrow{\overline{p_i}} G_i$ with $m_0^i = m_i \circ s_{i,L}$. It follows that $m_0^i = m_i \circ s_{i,L} = \tilde{m} \circ t_{i,L} \circ s_{i,L} = \tilde{m} \circ t_{0,L} = \tilde{m} \circ t_{j,L} \circ s_{j,L} = m_j \circ s_{j,L}$ and thus we have consistent matches with $m_0 := m_0^i$ well-defined and $G_0 = G_0^i$.



It remains to show the weakly independent matches. Given the above transformations we have pushouts (20_0), (20_i), (20_s) as above. Then we can find decompositions of (20_0) and (20_s) into pushouts $(27) + (28)$ and $(26) + (28)$, respectively. Using pushout (26) and Lemma A.8 it follows that (25) is a pushout, since $\tilde{K_s}$ is the colimit of $(s_{i,L})_{i=1,\ldots,n}$ and $\tilde{L}_0$ is the colimit of $(w_i)_{i=1,\ldots,n}$, and id_{K_0} is obviously an epimorphism.

Now Lemma A.6 implies that there is a decomposition into pushouts (24_i) with colimit D_0 of $(x_i)_{i=1,\ldots,n}$ and pushout (25_i) by $\mathcal{M}$-pushout pullback decomposition. Since D_0 is the colimit of $(x_i)_{i=1,\ldots,n}$ and (25_j) is a pushout it follows that D_j is the colimit of $(x_i)_{i=1,\ldots,j-1,j+1,\ldots,n}$ with morphisms $q_{ij} : P_i \to D_j$ and $d_{j0} \circ q_{ij} = y_{i0}$. Thus we obtain for all $i \neq j$ a morphism $p_{ij} = q_{ij} \circ x_{i0}$ and $f_j \circ p_{ij} = f_0 \circ d_{j0} \circ q_{ij} \circ x_{i0} = f_0 \circ y_{i0} \circ x_{i0} = m_i \circ u_i$.



3. Bijective Correspondence. Because of the uniqueness of the used constructions, the above constructions are inverse to each other up to isomorphism. □

Remark 3. Note, that q_i can be constructed as the amalgamated rule of the kernel morphisms $(p_{K_0} \to \overline{p_j})_{j \neq i}$, where $p_{K_0} = (K_0 \xleftarrow{id_{K_0}} K_0 \xrightarrow{id_{K_0}} K_0, \text{true}))$ and $\overline{p_j}$ is the complement rule of p_j.

For $n = 2$ and rules without application conditions, the Multi-Amalgamation Theorem specializes to the Amalgamation Theorem in [4]. Moreover, if p_0 is the empty rule, this is the Parallelism Theorem in [14], since the transformations are parallel independent for an empty kernel match.

Example 5. As already stated in Example 4, the transformations $G \xRightarrow{p_1,m_1} G_1$, $G \xRightarrow{p_2,m_2} G_2$, and $G \xRightarrow{p_1,m_3} G_3$ shown in Fig. 6 are s-amalgamable for the bundle $s = (s_1, s_2, s_3 = s_1)$ of kernel morphisms. Applying Fact 7, we can decompose these transformations into a transformation $G \xRightarrow{p_0,m_0} G_0$ followed by transformations $G_0 \xRightarrow{\overline{p_1},\overline{m_1}} G_1$, $G_0 \xRightarrow{\overline{p_2},\overline{m_2}} G_2$, and $G_0 \xRightarrow{\overline{p_1},\overline{m_3}} G_3$ via the complement rules, which are pairwise parallel independent. These transformations

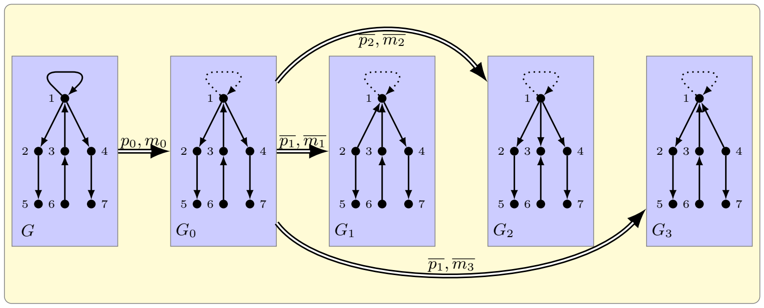


Fig. 7. The decomposition of the s-amalgamable bundle

are depicted in Fig. 7. Moreover, Thm. 2 implies that we obtain for this bundle of direct transformations an amalgamated transformation $G \xRightarrow{\tilde{p}_s, \tilde{m}} H$, which is the transformation already shown in Fig. 5. Vice versa, the analysis of this amalgamated transformation leads to the s-amalgamable bundle of transformations $G \xRightarrow{p_1, m_1} G_1$, $G \xRightarrow{p_2, m_2} G_2$, and $G \xRightarrow{p_1, m_3} G_3$ from Fig. 6.

Extension to Multi-Amalgamation with Maximal Matchings

An important extension of the presented theory is the introduction of interaction schemes and maximal matchings. An interaction scheme defines a bundle of kernel morphisms. In contrast to a concrete bundle, for the application of such an interaction scheme all possible matches for the multi rules are computed that agree on a given kernel match and lead to an amalgamable bundle of transformations. In our example, the interaction scheme $is = \{s_1, s_2\}$ contains the two kernel morphisms from Fig. 1. For the kernel match m_0, the matches m_1, m_2, m_3 are maximal: they are s-amalgamable, and any other match for p_1 or p_2 that agrees an m_0 would hold only already matched elements. This technique is very useful for the definition of the semantics of visual languages. For our example concerning statcharts [11], an unknown number of state transitions triggered by the same event, which is highly dependent on the actual system state, can be handled in parallel.

5 Conclusion

In this paper, we have generalized the theory of amalgamation in [4] to multi-amalgamation in $\mathcal{M}$-adhesive categories. More precisely, the Complement Rule and Amalgamation Theorems in [4] are presented on a set-theoretical basis for pairs of plain graph rules without any application conditions. The Complement Rule and Multi-Amalgamation Theorems in this paper are valid in adhesive and $\mathcal{M}$-adhesive categories for n rules with application conditions [13]. These generalizations are non-trivial and important for applications of parallel graph transformations to communication-based systems [8], to model transformations from BPMN to BPEL [15], and for the modeling of the operational semantics of visual languages [9], where interaction schemes are used to generate multi-amalgamated rules and transformations based on suitable maximal matchings.

The theory of multi-amalgamation is a solid mathematical basis to analyze interesting properties of the operational semantics, like termination, local confluence, and functional behavior. However, it is left open for future work to generalize the corresponding results in [2], like the Local Church–Rosser, Parallelism, and Local Confluence Theorems, to the case of multi-amalgamated rules, especially to the operational semantics of statecharts based on amalgamated graph transformation with maximal matchings in [11].

References

[1] Lack, S., Sobociński, P.: Adhesive Categories. In Walukiewicz, I., ed.: Proc. of FOSSACS 2004. Volume 2987 of LNCS., Springer (2004) 273–288

[2] Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs. Springer (2006)

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

[4] Böhm, P., Fonio, H.R., Habel, A.: Amalgamation of Graph Transformations: A Synchronization Mechanism. JCSS **34**(2-3) (1987) 377–408

[5] Ehrig, H., Kreowski, H.J.: Parallelism of Manipulations in Multidimensional Information Structures. In Mazurkiewicz, A., ed.: Proc. of MFCS'76. Volume 45 of LNCS., Springer (1976) 285–293

[6] Taentzer, G., Beyer, M.: Amalgamated Graph Transformations and Their Use for Specifying AGG - an Algebraic Graph Grammar System. In Schneider, H.J., Ehrig, H., eds.: Graph Transformations in Computer Science. Volume 776 of LNCS., Springer (1994) 380–394

[7] Heckel, R., Müller, J., Taentzer, G., Wagner, A.: Attributed Graph Transformations with Controlled Application of Rules. Technical Report B-19, Universitat de les Illes Balears (1995)

[8] Taentzer, G.: Parallel and Distributed Graph Transformation: Formal Description and Application to Communication Based Systems. PhD thesis, TU Berlin (1996)

[9] Ermel, C.: Simulation and Animation of Visual Languages based on Typed Algebraic Graph Transformation. PhD thesis, TU Berlin (2006)

[10] Löwe, M.: Algebraic Approach to Single-Pushout Graph Transformation. TCS **109** (1993) 181–224

[11] Golas, U., Biermann, E., Ehrig, H., Ermel, C.: A Visual Interpreter Semantics for Statecharts Based on Amalgamated Graph Transformation. (2010) submitted.

[12] Habel, A., Pennemann, K.H.: Correctness of High-Level Transformation Systems Relative to Nested Conditions. MSCS **19**(2) (2009) 245–296

[13] Ehrig, H., Habel, A., Lambers, L.: Parallelism and Concurrency Theorems for Rules with Nested Application Conditions. In Drewes, F., Habel, A., Hoffmann, B., Plump, D., eds.: Essays Dedicated to H.-J. Kreowski on the Occasion of His 60th Birthday. ECEASST (2010) 109–133 to appear.

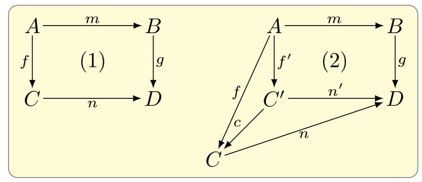
[14] Ehrig, H., Golas, U., Taentzer, G., Ermel, C., Biermann, E.: Parallel Independence of Amalgamated Graph Transformations Applied to Model Transformation. In

Engels, G., Lewerentz, C., Schäfer, W., Schürr, A., Westfechtel, B., eds.: Essays Dedicated to M. Nagl. LNCS, Springer (2010) to appear.

A Additional Lemmas

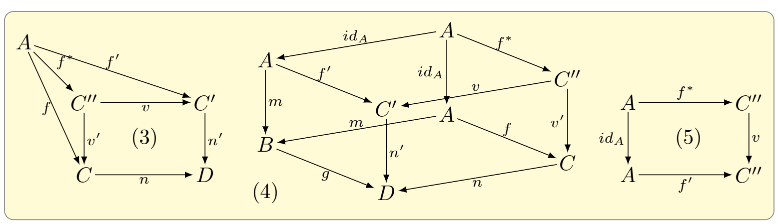
The following lemmas are valid in all adhesive and $\mathcal{M}$-adhesive categories and used in the proofs of the main theorems.

Lemma A.1. *If* (1) *is a pushout,* (2) *is a pullback, and* $n' \in \mathcal{M}$ *then there exists a unique morphism* $c : C' \to C$ *such that* $c \circ f' = f$, $n \circ c = n'$, *and* $c \in \mathcal{M}$.



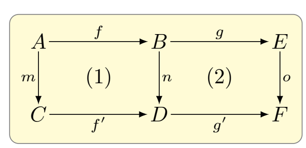
Proof. Since (2) is a pullback, $n' \in \mathcal{M}$ implies that $m \in \mathcal{M}$, and then also $n \in \mathcal{M}$ because (1) is a pushout.

Construct the pullback (3) with $v, v' \in \mathcal{M}$, and since $n' \circ f = g \circ m = n \circ f$ there is a unique morphism $f^* : A \to C''$ with $v \circ f^* = f'$ and $v' \circ f^* = f$. Now consider the following cube (4), where the bottom face is pushout (1), the back left face is a pullback because $m \in \mathcal{M}$, the front left face is pullback (2), and the front right face is pullback (3). Now by pullback composition and decomposition also the back right face is a pullback, and then the VK property implies that the top face is a pushout. Since (5) is a pushout and pushout objects are unique up to isomorphism this implies that v is an isomorphism and $C'' \cong C'$. Now define $c := v' \circ v^{-1}$ and we have that $c \circ f' = v' \circ v^{-1} \circ f' = v' \circ f^* = f$, $n \circ c = n \circ v' \circ v^{-1} = n'$, and $c \in \mathcal{M}$ by decomposition of $\mathcal{M}$-morphisms.



□

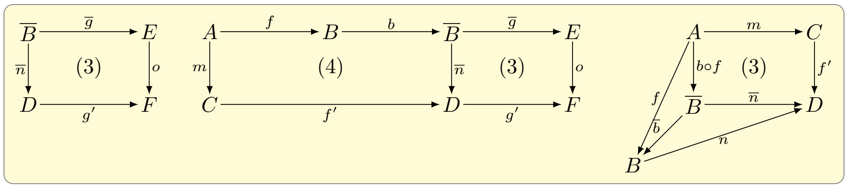
Lemma A.2. *If* (1) + (2) *is a pullback,* (1) *is a pushout,* (2) *commutes, and* $o \in \mathcal{M}$ *then also* (2) *is a pullback.*



Proof. With $o \in \mathcal{M}$, (1) + (2) being a pullback, and (1) being a pushout we have that $m, n \in \mathcal{M}$. Construct the pullback (3) of o and g', it follows that $\overline{n} \in \mathcal{M}$ and we get an induced morphism $b : B \to \overline{B}$ with $\overline{g} \circ b = g$, $\overline{n} \circ b = n$, and by decomposition of $\mathcal{M}$-morphisms $b \in \mathcal{M}$.

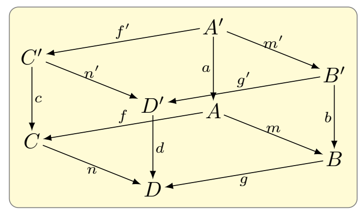
By pullback decomposition, also (4) is a pullback and we can apply Lemma 1 with pushout (1) and $\overline{n} \in \mathcal{M}$ to obtain a unique morphism $\overline{b} \in \mathcal{M}$ with $n \circ \overline{b} = \overline{n}$

and $\overline{b} \circ b \circ f = f$. Now $n \in \mathcal{M}$ and $n \circ \overline{b} \circ b = \overline{n} \circ b = n$ implies that $\overline{b} \circ b = id_B$, and similarly $\overline{n} \in \mathcal{M}$ and $\overline{n} \circ b \circ \overline{b} = n \circ \overline{b} = \overline{n}$ implies that $b \circ \overline{b} = id_{\overline{B}}$, which means that B and $\overline{B}$ are isomorphic such that also (2) is a pullback.

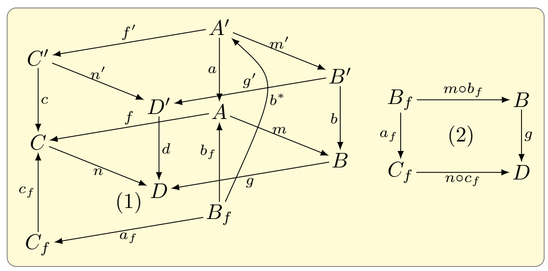


□

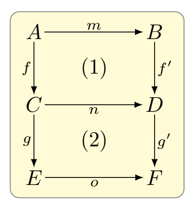
Lemma A.3. *Given the following commutative cube with the bottom face as a pushout, then the front right face has a pushout complement over $g \circ b$ if the back left face has a pushout complement over $f \circ a$.*



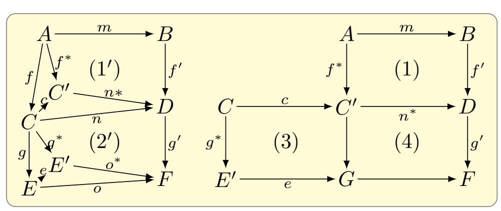
Proof. Construct the initial pushout (1) over f. Since the back left face has a pushout complement there is a morphism $b^* : B_f \to A'$ such that $a \circ b^* = b_f$. Since the bottom face is a pushout, (2) as the composition is the initial pushout over g. Now $b \circ m' \circ b^* = m \circ a \circ b^* = m \circ b_f$, and thus the pushout complement of $g \circ b$ exists. □



Lemma A.4. *Given pullbacks* (1) *and* (2) *with pushout complements over $f' \circ m$ and $g' \circ n$, respectively, then also* $(1)+(2)$ *has a pushout complement over $(g' \circ f') \circ m$.*

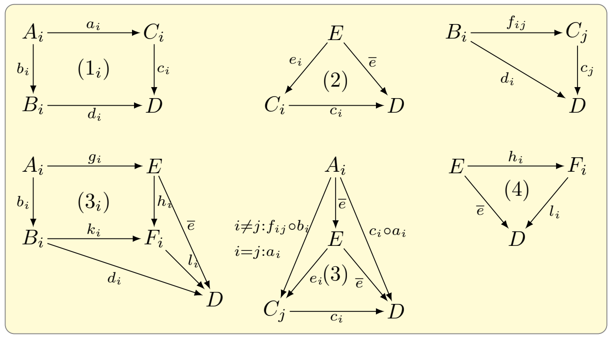


Proof. Let C' and E' be the pushout complements of (1) and (2), respectively. By Lemma 1 there are morphisms c and e such that $c \circ f = f^*$, $n^* \circ c = n$, $e \circ g = g*$, and $o^* \circ e = o$. Now $(2')$ can be decomposed into pushouts (3) and (4), and $(1')+(4)$ is also a pushout and the pushout complement of $(g' \circ f') \circ m$.



□

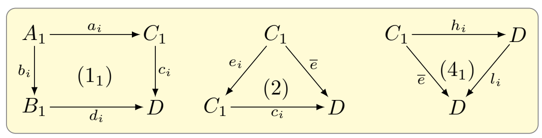
Lemma A.5. *Given the pushouts* (1_i) *and* (3_i) *with* $b_i \in \mathcal{M}$ *for* $i = 1, \ldots, n$*, morphisms* $f_{ij} : B_i \to C_j$ *with* $c_j \circ f_{ij} = d_i$ *for all* $i \neq j$*, and the limit* (2) *such that* g_i *is the induced morphism into* E *using* $c_j \circ f_{ij} \circ b_i = d_i \circ b_i = c_i \circ a_i$*, then* (4) *is the colimit of* $(h_i)_{i=1,\ldots,n}$*, where* l_i *is the induced morphism from pushout* (3_i) *compared with* $\overline{e} \circ g_i = c_i \circ e_i \circ g_i = c_i \circ a_i = d_i \circ b_i$*.*



Proof. We prove this by induction over n.

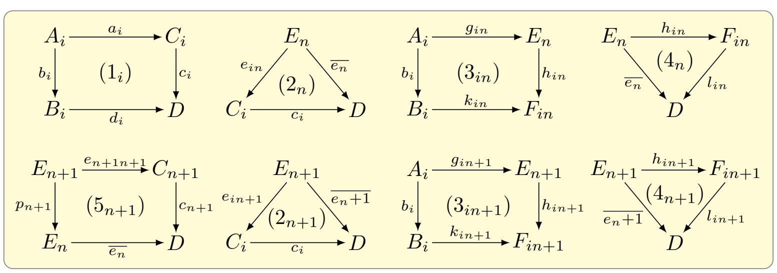
I.B. $n = 1$

For $n = 1$, we have that C_1 is the limit of c_1, i.e. $E = C_1$, it follows that $F_1 = C_1$ for the pushout $(3_1) = (1_1)$, and obviously (4_1) is a colimit.

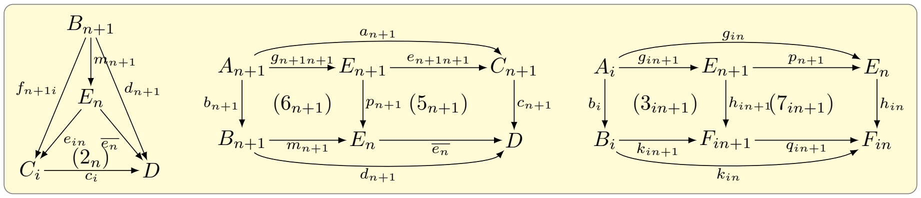


I.S. $n \to n + 1$

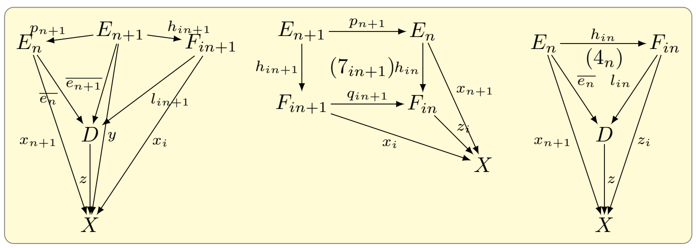
Consider the pushouts (1_i) with $b_i \in \mathcal{M}$ for $i = 1, \ldots, n+1$, morphisms $f_{ij} : B_i \to C_j$ with $c_j \circ f_{ij} = d_i$ for all $i \neq j$, the limits (2_n) and (2_{n+1}) of $(c_i)_{i=1,\ldots,n}$ and $(c_i)_{i=1,\ldots,n+1}$, respectively, leading to pullback (5_{n+1}) by construction of limits. Moreover, g_{in} and g_{in+1} are the induced morphisms into E_n and E_{n+1}, respectively, leading to pushouts (3_{in}) and (3_{in+1}). By induction hypothesis, (4_n) is the colimit of $(h_{in})_{i=1,\ldots,n}$, and we have to show that (4_{n+1}) is the colimit of $(h_{in+1})_{i=1,\ldots,n+1}$.



Since (2_n) is a limit and $c_i \circ f_{n+1i} = d_{n+1}$ for all $i = 1, \ldots, n$, we obtain a unique morphism m_{n+1} with $e_{in} \circ m_{n+1} = f_{n+1i}$ and $\overline{e_n} \circ m_{n+1} = d_{n+1}$. Since (1_{n+1}) is a pushout and (5_{n+1}) is a pullback, by $\mathcal{M}$-pushout-pullback decomposition also (5_{n+1}) and (6_{n+1}) are pushouts, and it follows that $F_{n+1n+1} = E_n$. From pushout $(3_{in+1}$ and $h_{in} \circ p_{n+1} \circ g_{in+1} = h_{in} \circ g_{in} = k_{in} \circ b_i$ we get an induced morphism q_{in+1} with $q_{in+1} \circ h_{in+1} = h_{in} \circ p_{n+1}$ and $q_{in+1} \circ k_{in+1} = kin$, and from pushout decomposition alsy (7_{in+1}) is a pushout.



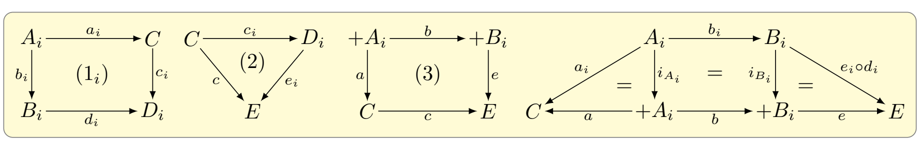
To show that (4_{n+1}) is a colimit, consider an object X and morphisms (x_i) and y with $x_i \circ h_{in+1} = y$ for $i = 1, \dots, n$ and $x_{n+1} \circ p_{n+1} = y$. From pushout (7_{in+1}) we obtain a unique morphism z_i with $z_i \circ q_{in+1} = x_i$ and $z_i \circ h_{in} = x_{n+1}$. Now colimit (4_n) induces a unique morphism z with $z \circ \overline{e_n} = x_{n+1}$ and $z \circ l_{in} = z_i$. It follows directly that $z \circ l_{in+1} = z \circ l_{in} \circ q_{in+1} = z_i \circ q_{in+1} = x_i$ and $z \circ \overline{e_{n+1}} = z \circ \overline{e_n} \circ p_{n+1} = x_{n+1} \circ p_{n+1} = y$. The uniqueness of z follows directly from the construction, thus (4_{n+1}) is the required colimit.



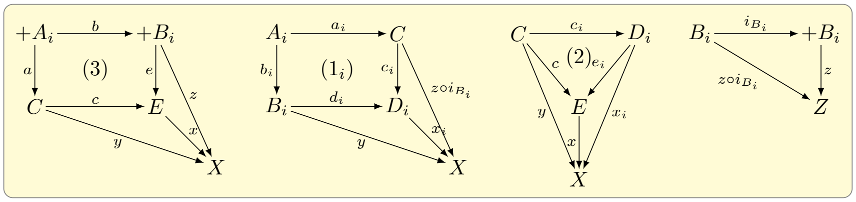
□

Lemma A.6. *Given the following diagrams* (1_i) *for* $i = 1, \dots, n$, (2), *and* (3), *with* $b = +b_i$, *and* a *and* e *induced by the coproducts* $+A_i$ *and* $+B_i$, *respectively, then we have:*

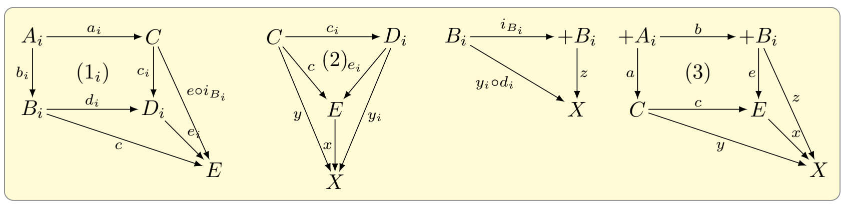
1. *If* (1_i) *is a pushout and* (2) *a colimit then also* (3) *is a pushout.*
2. *If* (3) *is a pushout then we find a decomposition into pushout* (1_i) *and colimit* (2) *with* $e_i \circ d_i = e \circ i_{B_i}$



Proof. 1. Given an object X and morphisms y, z with $y \circ a = z \circ b$. From pushout (1_i) we obtain with $z \circ i_{B_i} \circ b_i = z \circ b \circ i_{A_i} = y \circ a \circ i_{A_i} = y \circ a_i$ a unique morphism x_i with $x_i \circ c_i = y$ and $x_i \circ d_i = z \circ i_{B_i}$. Now colimit (2) implies a unique morphism x with $x \circ c = y$ and $x \circ e_i = x_i$. It follows that $x \circ e \circ i_{B_i} = x \circ e_i \circ d_i = x_i \circ d_i = z \circ i_{B_i}$, and since z is unique w.r.t. $z \circ i_{B_i}$ it follows from the coproduct that $z = x \circ e$. Uniqueness of x follows from the uniqueness of x and x_i, and hence (3) is a pushout.

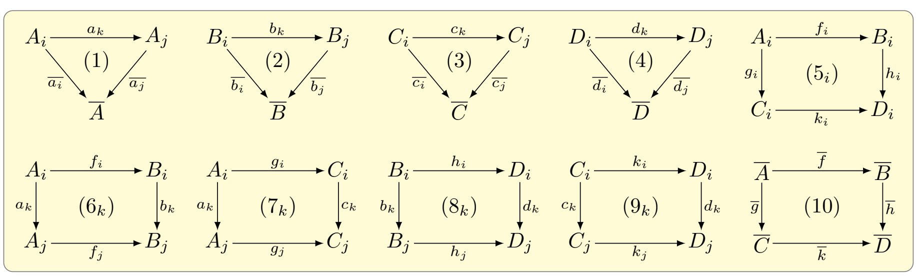


2. Define $a_i := a \circ i_{A_i}$. Now construct pushout (1_i). With $e \circ i_{B_i} \circ b_i = e \circ b \circ i_{A_i} = c \circ a_i$ pushout (1_i) induces a unique morphism e_i with $e_i \circ d_i = e \circ i_{B_i}$ and $e_i \circ c_i = c$. Given an object X and morphisms y, y_i with $y_i \circ c_i = y$ we obtain a morphism z with $z \circ i_{B_i} = y_i \circ d_i$ from coproduct $+B_i$. Then we have that $y \circ a \circ i_{A_i} = y_i \circ c_i \circ a_i = y_i \circ d_i \circ b_i = z \circ i_{B_i} \circ b_i = z \circ b \circ i_{A_i}$, and from coproduct $+A_i$ it follows that $y \circ a = z \circ b$. Now pushout (3) implies a unique morphism x with $x \circ c = y$ and $x \circ e = z$. From pushout (1_i) using $x \circ e_i \circ d_i = x \circ e \circ i_{B_i} = z \circ i_{B_i} = y_i \circ d_i$ and $x \circ e_i \circ c_i = x \circ c = y = y_i \circ c_i$ it follows that $x \circ e_i = y_i$, thus (2) is a colimit.



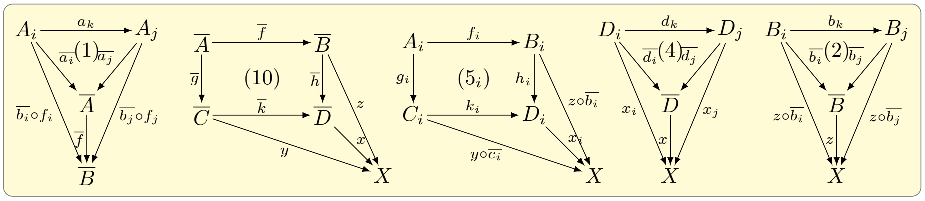
□

Lemma A.7. *Consider colimits* (1) – (4) *such that* (5_i) *is a pushout for all* $i = 1, \dots, n$ *and* (7_k) – (9_k) *commute for all* $k = 1, \dots, m$*. Then also* (10) *is a pushout.*



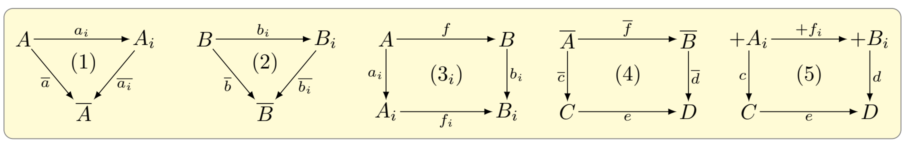
Proof. The morphisms $\overline{f}$, $\overline{g}$, $\overline{h}$, and $\overline{k}$ are uniquely induced by the colimits. We show this examplarily for the morphism $\overline{f}$: From colimit (1), with $\overline{b_j} \circ f_j \circ a_k = \overline{b_j} \circ b_k \circ f_i = \overline{b_i} \circ f_i$ we obtain a unique morphism $\overline{f}$ with $\overline{f} \circ \overline{a_i} = \overline{b_i} \circ f_i$. It follows directly that $\overline{k} \circ \overline{h} = \overline{h} \circ \overline{f}$.

Now consider an object X and morphisms y, z with $y \circ \overline{g} = z \circ \overline{f}$. From pushout (5_i) with $y \circ \overline{c_i} \circ g_i = y \circ \overline{g} \circ \overline{a_i} = z \circ \overline{f} \circ \overline{a_i} = z \circ \overline{b_i} \circ f_i$ we obtain a unique morphism x_i with $x_i \circ k_i = y \circ \overline{c_i}$ and $x_i \circ h_i = z \circ \overline{b_i}$.

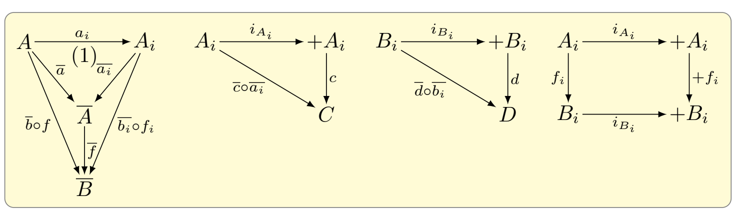


For all $k = 1, \ldots, m$, $x_j \circ d_k \circ k_i = x_j \circ k_j \circ c_k = y \circ \overline{c_j} \circ c_k = y \circ \overline{c_i}$ and $x_j \circ d_k \circ h_i = x_j \circ h_j \circ b_k = z \circ \overline{b_j} \circ b_k = z \circ \overline{b_i}$, and pushout (5_i) implies that $x_i = x_j \circ d_k$. This means that colimit (4) implies a unique x with $x \circ \overline{d_i} = x_i$. Now consider colimit (2), and $x \circ \overline{h} \circ \overline{b_i} = x \circ \overline{d_i} \circ h_i = x_i \circ h_i = z \circ \overline{b_i}$ implies that $x \circ \overline{h} = z$. Similarly, $x \circ \overline{k} = y$, and the uniqueness follows from the uniqueness of x with respect to (4). Thus, (10) is indeed a pushout. □

Lemma A.8. *Consider colimits* (1) *and* (2) *such that* (3_i) *commutes for all* $i = 1, \ldots, n$, f *is an epimorphism, and* (4) *is a pushout with* $\overline{f}$ *induced by colimit* (1). *Then also* (5) *is a pushout, where* c *and* d *are induced from the coproducts.*

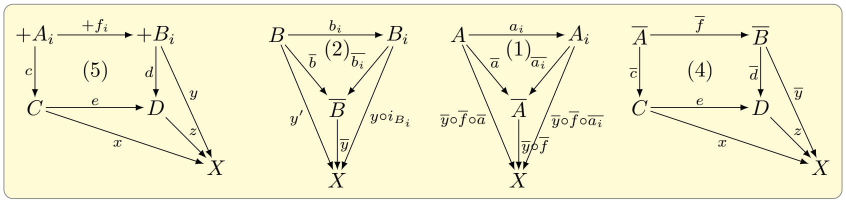


Proof. Since (1) is a colimit and $\overline{b_i} \circ f_i \circ a_i = \overline{b_i} \circ b_i \circ f = \overline{b} \circ f$, we actually get an induced $\overline{f}$ with $\overline{f} \circ \overline{a_i} = \overline{b_i} \circ f_i$ and $\overline{f} \circ \overline{a} = \overline{b} \circ f$. From the coproducts, we obtain induced morphisms c with $c \circ i_{A_i} = \overline{c} \circ \overline{a_i}$ and d with $d \circ i_{B_i} = \overline{d} \circ \overline{b_i}$. Moreover, for all $i = 1, \ldots, n$ we have that $d \circ (+f_i) \circ i_{A_i} = d \circ i_{B_i} \circ f_i = \overline{d} \circ \overline{b_i} \circ f_i = \overline{d} \circ \overline{f} \circ \overline{a_i} = e \circ \overline{c} \circ \overline{a_i} = e \circ c \circ i_{A_i}$. Uniqueness of the induced coproduct morphisms leads to $d \circ (+f_i) = e \circ c$, i.e. (5) commutes.



We have to show that (5) is a pushout. Given morphisms x, y with $x \circ c = y \circ (+f_i)$, we have that $y \circ i_{B_i} \circ b_i \circ f = y \circ i_{B_i} \circ f_i \circ a_i = y \circ (+f_i) \circ i_{A_i} \circ a_i = x \circ c \circ i_{A_i} \circ a_i = x \circ \overline{c} \circ \overline{a_i} \circ a_i = x \circ \overline{c} \circ \overline{a}$ for all $i = 1, \ldots, n$. f being an epimorphisms implies that $y \circ i_{B_i} \circ b_i = y \circ i_{B_j} \circ b_j$ for all i, j. Now define $y' := y \circ i_{B_i} \circ b_i$, and from colimit (2) we obtain a unique morphism $\overline{y}$ with $\overline{y} \circ \overline{b_i} = y \circ i_{B_i}$ and $\overline{y} \circ \overline{b} = y'$.

Now $x \circ \overline{c} \circ \overline{a_i} = x \circ c \circ i_{A_i} = y \circ (+f_i) \circ i_{A_i} = y \circ i_{B_i} \circ f_i = \overline{y} \circ \overline{b_i} \circ f_i = \overline{y} \circ \overline{f} \circ \overline{a_i}$ and $x \circ \overline{c} \circ \overline{a} = x \circ \overline{c} \circ \overline{a_i} \circ a_i = \overline{y} \circ \overline{f} \circ \bar{\imath} \circ a_i = \overline{y} \circ \overline{f} \circ \overline{a}$, and the uniqueness of the induced colimit morphism implies that $\overline{y} \circ \overline{f} = x \circ \overline{c}$. This means that X can be compared to pushout (4), and we obtain a unique morphism z with $z \circ \overline{d} = \overline{y}$ and $z \circ e = x$. Now $z \circ d \circ i_{B_i} = z \circ \overline{d} \circ \overline{b_i} = \overline{y} \circ \overline{b_i} = y \circ i_{B_i}$, and it follows that $z \circ d = y$. Similarly, the uniqueness of z w.r.t. the pushout propert of (5) follows, thus (5) is a pushout.



□