Graph Transformations

An Introduction
to the
Categorical Approach

Hans J. Schneider
Chapter 4

Effectively Constructing Derivation Steps

In Definition 3.1.2, we have introduced the notion of derivability with respect to arbitrary categories using the following symmetrical diagram:

\[
\begin{array}{c}
B^l \xleftarrow{p^l} I \xrightarrow{p^r} B^r \\
g^l \downarrow \quad PO \quad g \quad PO \quad g^r \\
G^l \xleftarrow{p^l} C \xrightarrow{p^r} G^r
\end{array}
\]

\(G^l\) and \(G^r\) are unambiguously defined if we know the production \(p = (p^l, p^r)\) and the embedding \(g\). Usually, however, the embedding \(g\) is not given, but the handle \(g^l: B^l \rightarrow G^l\):

\[
\begin{array}{c}
B^l \xleftarrow{p^l} I \xrightarrow{p^r} B^r \\
g^l \downarrow \\
G^l
\end{array}
\]

and we have to look for a suitable context object \(C\) that allows an embedding \(g: I \rightarrow C\) such that the given \(G^l\) is the pushout object of \(g\) and \(p^l\). If we have found such an embedding, the right-hand part of the derivability diagram is unambiguously defined, and we can construct it in a straightforward manner.

By this reason, the main problem in constructing derivation steps is completing a pushout diagram backwards. As Rosen has outlined [75], the problem is decidable in the most interesting case, namely in the case of finite graphs. He gives upper bounds to the number of nodes and to the number of edges of the context graph \(C\). Therefore, only a finite number of graphs has to be taken into consideration, and these graphs can be tested one after the other.
In this chapter, however, we are interested in effectively constructing the context object, i.e., we are looking for a procedure straightforwardly directed to the solution. For this purpose, we show that the problem can be simplified by considering epimorphisms and monomorphisms separately. First, we describe the construction in \( \text{Set} \) for nodes and edges. Then, we examine how to combine these solutions. Finally, we take labeled graphs into consideration.

### 4.1 Pushout Complements and Their Decomposition

As we have already mentioned, the main problem in constructing derivation steps effectively is completing a pushout diagram backwards:

**Definition 4.1.1** (Pushout complement):
If the diagram \((a)\) can be completed such that diagram \((b)\) is a pushout diagram:

\[
\begin{array}{ccc}
(a) & \stackrel{p}{\longrightarrow} & B \\
\downarrow{\bar{g}} & & \downarrow{\bar{g}} \\
G & & C \\
\end{array}
\begin{array}{ccc}
(b) & \stackrel{p}{\rightarrow} & B \\
\downarrow{g} & & \downarrow{\bar{g}} \\
 & & G \\
\end{array}
\]

then, we call \( C \) together with morphisms \( g \) and \( \bar{p} \) a pushout complement of \( \bar{g} \cdot p \).

For reason of simplicity, we often call \( C \) the pushout complement without explicitly referring to the morphisms.

Example 3.2.4 shows that the pushout complement need not be uniquely defined. On the other hand, there are examples which can not be completed at all, e.g., in \( \text{Set} \), we can not find a pushout complement if \( \bar{g} \) does not satisfy the identification condition:

**Lemma 4.1.2** (Identification condition, Rosen [75]):
If in \( \text{Set} \), a pushout complement of \( \bar{g} \cdot p \) exists, then \( \bar{g} \) satisfies:

\[
\bar{g}(x) = \bar{g}(x') \implies x = x' \lor x, x' \in p[I]
\]

This lemma is an immediate consequence of Corollary 2.5.9. We illustrate it by a small example:

\[
\begin{array}{ccc}
\{1, 3\} & \stackrel{p}{\longrightarrow} & \{1, 3, 4, 5\} \\
\downarrow{\bar{g}} & & \downarrow{\bar{g}} \\
\{1, [3, 4], 5\} & & \\
\end{array}
\]

The identification condition is not satisfied: Elements 3 and 4 are put together by \( \bar{g} \). Since a pushout diagram is commutative, these elements must also be put together along the complementary path \( \bar{p} \cdot g \). But this is impossible because 4 has no pre-image in \( I \).
A first observation concerning pushout complements follows from the preservation theorem (Theorem 2.5.1):

**Corollary 4.1.3:**
Constructing a pushout complement, we have the following relations between the morphisms:

(a) If \( p \) is an epimorphism (a coretraction), then \( \bar{p} \) is an epimorphism (a coretraction), too.

(b) Only if the given \( \bar{g} \) is an epimorphism (a coretraction), \( g \) can be an epimorphism (a coretraction).

Please note that the preservation theorem does not allow us to deduce that \( g \) must be an epimorphism or a coretraction if \( \bar{g} \) is. In the following example in \( \text{Set} \), \( \bar{g} \) is the identity, but \( g \) is not even injective:

\[
\begin{align*}
\{1, 2, 3, 4, 5, 6\} & \xrightarrow{p} \{[1, 2], [3, 4], [5, 6]\} \\
\{[1, 2], [3, 4], [5, 6]\} & \xrightarrow{\bar{p}} \{[1, 2], [3, 4], 5, 6\}
\end{align*}
\]

Nevertheless, the corollary suggests that looking for a pushout complement may become simpler if we restrict discussion to classes of morphisms that are preserved by the pushout construction. Again, considering \( \text{Set} \) and \( \text{Graph} \) can give us an idea how to proceed. In these categories, we can split each morphism into two morphisms the first of which is surjective and the second is injective.

**Example 4.1.4:**
We consider a simple example in \( \text{Graph} \):

\[
\begin{align*}
\begin{array}{c}
1 \\
5 \\
3
\end{array} & \overset{\text{e}}{\xrightarrow{\text{f}}} \\
\begin{array}{c}
6 & 2 & 7 \\
7 & 2 & 7 \\
3 & 4 & 2
\end{array}
\end{align*}
\]

Splitting this graph morphism into a surjective part and an injective one yields the following decomposition:

\[
\begin{align*}
\begin{array}{c}
1 \\
5 \\
3
\end{array} & \overset{\text{e}}{\xrightarrow{\text{f}}} \\
\begin{array}{c}
6 & 2 & 7 \\
7 & 2 & 7 \\
3 & 4 & 2
\end{array}
\end{align*}
\]
When constructing pushout diagrams, we can take advantage of this decomposition: In the first step, we concentrate upon putting elements together and in the second, we add some elements.

The following definition generalizes this observation by introducing categories that allow such a decomposition:

**Definition 4.1.5 (E−M-factorizable category):**

Given a category \( K \), let \( E \) be a class of epimorphisms that contains all the isomorphisms of \( K \) and is closed under composition, and let \( M \) be a class of monomorphisms that contains all the isomorphisms of \( K \) and is closed under composition. Then, \( K \) is called \( E−M \)-factorizable if and only if we can split each morphism \( f \in \text{Mor}_K \) unambiguously (up to ismorphism) into two morphisms such that

\[
(\exists e \in E)(\exists m \in M)(f = m \cdot e)
\]

Please note that \( E \) and \( M \) are not disjoint because both contain the isomorphisms, and that we do not require \( E \) to contain all the epimorphisms and \( M \) to contain all the monomorphisms.

**Example 4.1.6:**

The category \( \text{Set} \) is \( E−M \)-factorizable if we choose the set of all surjections as \( E \) and the set of all injections as \( M \). The same holds true in the category \( \text{Graph} \), since we can factorize \((f_E, f_V)\) component by component and we can make \((m_E, m_V)\) and \((e_E, e_V)\) graph morphisms (Exercise 4.8.2).

Again, \( \text{Setincl} \) is an example of a category with rather an interesting behavior:

**Example 4.1.7:**

The category \( \text{Setincl} \) is \( E−M \)-factorizable if we choose the set of all morphisms as \( E \) and the set of isomorphisms as \( M \) (or vice versa). Since each morphism of this category is both an epimorphism and a monomorphism, the decomposition is trivial.

We now consider a pushout diagram in an \( E−M \)-factorizable category. It is possible to split the morphisms \( p, g, \bar{p}, \) and \( \bar{g} \) into epimorphic and monomorphic components as shown in the middle:

\[\begin{array}{c}
g \\
p \\ PO \\
g_m \\
p_e \\
p_m \\
 \bar{g} \\
g_e \\
gm_e \\
\bar{p}_m \\
\bar{p}_e \\
p \\
\end{array}\]

Our aim is to decompose the pushout diagram into four subdiagrams each of which is a pushout diagram. Obviously, the direction from right to left is an immediate consequence of the lemma on composing pushouts. (Lemma 2.5.18). If we can establish
the decomposition from left to right, all the morphisms with index \( e \) are epimorphic by the preservation theorem (Theorem 2.5.1). But, what about the other morphisms? The preservation theorem states that in an arbitrary category, coretractions are preserved, but monomorphisms need not: We can say that all these morphisms are in \( \mathcal{M} \) if we restrict this class to coretractions, but this restriction would exclude many interesting examples in the category \( \text{Graph} \). A closer look to the following proof shows that we can weaken the assumptions. We need only the preservation property:

**Theorem 4.1.8** (Decomposition of pushouts):

Let \( \mathcal{K} \) be an \( \mathcal{E} - \mathcal{M} \)-factorizable category that has pushouts such that \( \mathcal{E} \) and \( \mathcal{M} \) are closed under construction of pushouts, i.e., if a given morphism is in \( \mathcal{E} \) (in \( \mathcal{M} \)), then the morphism constructed on the opposite side of the diagram is in \( \mathcal{E} \) (in \( \mathcal{M} \)), too. Then, each pushout diagram \( \bar{g} \cdot p = \bar{p} \cdot g \) in \( \mathcal{K} \) can be unambiguously split into two pushout diagrams such that \( g = g_m \cdot g_e \) and \( \bar{g} = \bar{g}_m \cdot \bar{g}_e \) where \( g_e, \bar{g}_e \in \mathcal{E}, g_m, \bar{g}_m \in \mathcal{M} \):

![Diagram of pushouts](image)

**Proof:**

We consider the pushout diagram on the left-hand side. We can decompose \( g \) into \( g = g_m \cdot g_e \). Then, we construct the pushout diagram (1) of \( p \) and \( g_e \):

![Diagram of pushout (1)](image)

\( e' \) is in \( \mathcal{E} \) by the assumption. Next, we construct the pushout diagram (2) of \( p' \) and \( g_m \), and we get the morphisms \( \bar{p}' \) and \( m' \). The latter is in \( \mathcal{M} \). By the lemma on composing pushouts, we can put (1) and (2) together. This yields a pushout of \( p \) and \( g_m \cdot g_e = g \). Since pushouts are unambiguous up to isomorphisms, the diagram must be isomorphic to the original pushout diagram, and we have \( \bar{p} = \bar{p}' \) and \( \bar{g} = m' \cdot e' \). This completes the proof from left to right. The converse immediately follows from Lemma 2.5.18.

\( \square \)

Applying this theorem twice, first decomposing \( g \) and then decomposing \( p \), we get what we want:
Corollary 4.1.9 (\(\mathcal{E}-\mathcal{M}\)-Decomposition of pushouts):

Under the assumptions of Theorem 4.1.8, we can split each pushout diagram into four pushout diagrams such that each morphism with index \(e\) is in \(\mathcal{E}\) and each morphism with index \(m\) is in \(\mathcal{M}\):

\[
\begin{array}{c}
g \quad PO \quad \tilde{g} \\
p & \downarrow & \\
p_e & & g_e & \quad PO & \quad g_e' & \quad PO & \quad \tilde{g}_e \\
p_m & & g_m & \quad PO & \quad g_m' & \quad PO & \quad \tilde{g}_m \\
\end{array}
\]

Usually, you find the theorem in the literature as follows:

Corollary 4.1.10:

Let \(\mathcal{K}\) be an \(\mathcal{E}-\mathcal{M}\)-factorizable category with \(\mathcal{E}\) being the set of all epimorphisms and \(\mathcal{M}\) being the set of all coretractions. Then, each pushout diagram in \(\mathcal{K}\) can be split into four pushout diagrams as shown in the previous corollary.

In the category of graphs, coretractions are too much a restriction as we have discussed in Example 2.3.16. Therefore, we have formulated Theorem 4.1.8 without referring to coretractions. This makes it suitable for our purposes. Although we discuss the situation in \(\text{Graph}\) in detail not until Section 4.5, let us consider a first example of decomposing graph morphisms:

Example 4.1.11:

We consider the following morphisms \(p\) and \(g\):

\[
\begin{array}{c}
\begin{array}{c}
1 \\
8 \\
3 \\
\end{array}
\end{array}
\quad \xrightarrow{p} \quad
\begin{array}{c}
\begin{array}{c}
1 \quad 2 \\
8 \quad [9, 10] \\
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
5 \\
12 \\
6 \\
\end{array}
\end{array}
\quad \xrightarrow{g} \\n\rightdownarrow
\]

\[
\begin{array}{c}
\begin{array}{c}
11 \quad [1, 2] \\
9 \quad 10 \\
8 \quad \quad 13 \\
3 \\
\end{array}
\end{array}
\quad \xrightarrow{p} \\
\end{array}
\]

As usual, the mappings are indicated by the numbers. Nodes 1 and 2 are put together by \(g\), and nodes 3 and 4 as well as edges 9 and 10 are put together by \(p\). We decompose these morphisms into surjections and injections, and then we construct the pushout diagram concentrating on either putting elements together or adding elements. To simplify the figure (on top of the next page), we omit the edge numbers.
Example 4.1.11: Constructing the pushout diagram step by step

Concentrating on subclasses of morphisms that have special properties simplifies constructing the pushout if you do it by hand as well as if you design a computer program. We can take advantage of this decomposition even more if we construct the pushout complement. We decompose the given morphisms \( p \) and \( \bar{g} \) into \( p = p_m \cdot p_e \) and \( \bar{g} = \bar{g}_m \cdot \bar{g}_e \), respectively, as shown on the left-hand side:

In the first step, we construct the pushout complement of a monomorphism and an epimorphism. In the second step, we have two monomorphisms, in the third, two epimorphisms.\(^1\) In the last step, we have to deal with an epimorphism and a monomorphism as in the first step, but the situation is inverted.

\(^1\)Of course, you can exchange step (2) and step (3).
In Chapter 3, we have constructed pushout complements in the categories of sets and graphs, intuitively. We have built the context object by removing the elements of the left-hand object and adding the elements of the interface object. In terms of set theory, the first step is constructing the complementary set of $B$ with respect to $G$, and the second is constructing the disjoint union. The disjoint union is the coproduct in $\text{Set}$. Therefore, a reasonable generalization may be introducing the notion of a coproduct complement:

**Definition 4.2.1 (Coproduct complement):**

The morphism $\bar{f} : \bar{A} \to B$ is called a coproduct complement of $f : A \to B$ if and only if $A \xrightarrow{f} B \xleftarrow{\bar{f}} \bar{A}$ is a coproduct.

In $\text{Set}$, the situation is simple: If a morphism is injective, it has a unique coproduct complement, otherwise it has no coproduct complement at all:

**Example 4.2.2 (Coproduct complement in $\text{Set}$):**

In $\text{Set}$, a morphism has a unique coproduct complement if and only if it is injective. By lemma 2.4.3, both morphisms in a coproduct are injective. Conversely, if we have an injection $f : A \to B$, we can construct the complementary set $\bar{A} = B \setminus f[A]$. Then, $\bar{f}$ is the natural injection $\bar{A} \to B$.

In $\text{Set}_{incl}$, we have a different situation:

**Example 4.2.3 (Coproduct complement in $\text{Set}_{incl}$):**

As we have seen in Example 2.5.13, the coproduct in $\text{Set}_{incl}$ is the set-theoretic union. Therefore, each morphism, i.e., each inclusion, has a coproduct complement, but this coproduct complement need not be unambiguous. Consider $A = \{1, 2\}$ and $B = \{1, 2, 3\}$. We have four coproduct complements given by the objects $\bar{A}_1 = \{3\}$, $\bar{A}_2 = \{1, 3\}$, $\bar{A}_3 = \{2, 3\}$, $\bar{A}_4 = \{1, 2, 3\}$ together with the appropriate inclusions.

Now, we can formulate the procedure mentioned above in a more general setting: First, construct a coproduct complement of the handle; then, construct the coproduct of this object and of the interface object to get the context object. This construction, due to Ehrig and Kreowski [29], is based on the following two lemmata:

**Lemma 4.2.4 (Ehrig/Kreowski [29]):**

If the diagram

\[
\begin{array}{ccc}
I & \xrightarrow{p} & B \\
\downarrow{g} & = & \downarrow{\bar{g}} \\
C & \xleftarrow{\bar{p}} & G
\end{array}
\]

is commutative and if $g$ has a coproduct complement $g'$, then the diagram is a pushout diagram if and only if $(\bar{g}, \bar{p} \cdot g')$ is a coproduct.
Proof:

We consider the following diagram with \((g,g')\) being a coproduct. First, we assume that \((1)\) is a pushout diagram. We have to show that \((\bar{g},\bar{p} \cdot g')\) is a coproduct, i.e., for any pair \(f_1: B \to X\) and \(f_2: I' \to X\), we can find a unique \(f: G \to X\) such that \(f_1 = f \cdot \bar{g}\) and \(f_2 = f \cdot \bar{p} \cdot g'\) hold. Since \((g,g')\) is a coproduct, there is a unique \(f_3: C \to X\) with \(f_1 \cdot p = f_3 \cdot g\) and \(f_2 = f_3 \cdot g'\). The first of these equations allows us to apply the pushout property of diagram \((1)\): There exists a unique \(f: G \to X\) with \(f_1 = f \cdot \bar{g}\) and \(f_3 = f \cdot \bar{p} \cdot g'\). Thus, we have found a suitable \(f\). Although each step yields an unambiguous morphism, we have to make sure that there is no other way to find a different \(f'\). In this case, however, the pushout property of diagram \((1)\) would show that these morphisms are equal up to isomorphisms.

Conversely, we assume that \((\bar{g},\bar{p} \cdot g')\) is a coproduct and we show the pushout property of \((1)\). We start with two morphisms \(f_1\) and \(f_3\) satisfying \(f_1 \cdot p = f_3 \cdot g\), and we define the morphism \(f_2 := f_3 \cdot g'\). Since \((\bar{g},\bar{p} \cdot g')\) is a coproduct, there exists a unique \(f: G \to X\) such that \(f_1 = f \cdot \bar{g}\) and \(f_2 = f \cdot \bar{p} \cdot g'\). Whereas the first equation is already part of what we are looking for, the second can be used to factorize \(f_3\) taking advantage of \((g,g')\) being a coproduct. We consider \(f_1 : I \to X\) and \(f_2 : I' \to X\), and we know that there is a unique \(f_3: C \to X\) with \(f_2 = f_3 \cdot g'\) and \(f_1 \cdot p = f_3 \cdot g\). Since \(f_3\) and \(f \cdot \bar{p}\) also satisfy these equations (e.g., \(f \cdot \bar{p} \cdot g = f \cdot \bar{g} \cdot p = f_1 \cdot p\)), the morphisms \(f_3, f_3,\) and \(f \cdot \bar{p}\) must be equal, i.e., \(f_3 = f \cdot \bar{p}\). □

This lemma is not constructive since it assumes existence of the context object \(C\), but the next is:

**Lemma 4.2.5** (Ehrig/Kreowski \[29\]):

If \(p: I \to B\) is an arbitrary morphism and if \(\bar{g}: B \to G\) has a coproduct complement, then an object \(C\) and two morphisms \(g\) und \(\bar{p}\) exist such that the following diagram is a pushout diagram:

```
\[
\begin{array}{ccc}
I & \xrightarrow{p} & B \\
\downarrow{g} & & \downarrow{\bar{g}} \\
C & \xrightarrow{\bar{p}} & G
\end{array}
\]
```

Proof:

We consider a coproduct complement \(\bar{g}' : B' \to G\) of \(\bar{g}\), and we construct \(C\) together
with morphisms $g$ and $g'$ as the coproduct of $I$ and $B'$. From this, we get a unique \( \bar{p} : C \to G \) with \( \bar{g}' = \bar{p} \cdot g' \) and \( \bar{g} \cdot p = \bar{p} \cdot g \). The last equation implies that diagram (1) is commutative. Since both \((g, g')\) and \((\bar{g}, \bar{g}')\) are coproducts, the pushout property immediately follows from the previous lemma.

We can use this proof to construct a pushout complement in the case that a coproduct complement of \( \bar{g} \) exists.

**Example 4.2.6:**
We consider a simple example in $\textbf{Set}$:

\[
\begin{array}{ccc}
I : \{a_1, b_1\} & \xrightarrow{p \colon x_1 \mapsto x_2} & B : \{a_2, b_2, c_2\} \\
\downarrow g_1 & & \downarrow \bar{g} \\
C & \xrightarrow{\bar{p} \colon x_2 \mapsto x_3} & G : \{a_3, b_3, c_3, d_3, e_3\}
\end{array}
\]

The morphisms are indicated by the indices, e.g., \( p(a_1) = a_2 \). \( B' \) is easily constructed by taking all the elements of $G$ that are not images under $\bar{g}$, i.e., $d_3, e_3$, and defining a mapping with these elements as images:

\[
\begin{array}{ccc}
I : \{a_1, b_1\} & \xrightarrow{p \colon x_1 \mapsto x_2} & B : \{a_2, b_2, c_2\} \\
\downarrow g_1 & & \downarrow \bar{g} \\
C & \xrightarrow{\bar{p} \colon x_2 \mapsto x_3} & G : \{a_3, b_3, c_3, d_3, e_3\} \\
\downarrow \bar{g}' & & \downarrow \bar{g}' \\
B' : \{d_4, e_4\} & & \{d_4, e_4\}
\end{array}
\]

Finally, we construct the coproduct of $I$ and $B'$:

\[
\begin{array}{ccc}
I : \{a_1, b_1\} & \xrightarrow{p \colon x_1 \mapsto x_2} & B : \{a_2, b_2, c_2\} \\
g : x_1 \mapsto x_5 & & \bar{g} : x_2 \mapsto x_3 \\
\downarrow C : \{a_5, b_5, d_5, e_5\} & \xrightarrow{\bar{p} \colon x_5 \mapsto x_3} & G : \{a_3, b_3, c_3, d_3, e_3\} \\
g' : x_4 \mapsto x_5 & & \bar{g}' : x_4 \mapsto x_3 \\
B' : \{d_4, e_4\} & & \{d_4, e_4\}
\end{array}
\]
Then, $\bar{p}$ is defined by the coproduct property. The reader can easily verify that $\bar{g} \cdot p = \bar{p} \cdot g$ is a pushout diagram.

The proof of Lemma 4.2.5 shows that each coproduct complement of $\bar{g}$ leads to a pushout complement. In $\mathcal{S}et$, a morphism has a coproduct complement if and only if it is an injection, and the coproduct complement is unambiguous up to isomorphisms. Since the coproduct construction also is unambiguous, we can summarize the situation in $\mathcal{S}et$ as follows:

**Corollary 4.2.7:**

In $\mathcal{S}et$, we have: If $\bar{g}$ is an injection, then there exists exactly one pushout complement such that $g$ is an injection.

Why is it impossible to find another pushout complement with an injective $g$? Since $I$ is given and each element of $I$ must have an unambiguous image in $C$, only the additional elements of $C$ may be chosen differently, i.e., another $g$ would result in a different $B'$ and therefore, in a different $G$, but $G$ is given.

Although we can apply the algorithm of Lemma 4.2.5 to construct a pushout complement in the case of an injective handle, it does not yield all possible solutions, since the coproduct construction does not allow a noninjective $g$, but there may exist noninjective solutions:

**Example 4.2.8:**

We modify the morphism $p$ such that $p(a_1) = p(b_1) = [a_2, b_2]$. Of course, we must change $\bar{g}$, too. As before, we can construct the coproduct complement $\bar{g}' : B' \to G$ and the coproduct $(g, g')$. We sum up this construction in the following figure:

\[
\begin{array}{c}
I : \{a_1, b_1\} \\
g : x_1 \mapsto x_5 \\
C_1 : \{a_5, b_5, d_5, e_5\} \\
g' : x_4 \mapsto x_5 \\
B' : \{d_4, e_4\}
\end{array}
\begin{array}{c}
p : x_1 \mapsto x_2 \\
\bar{p} : x_5 \mapsto x_3 \\
G : \{[a_3, b_3], c_3, d_3, e_3\}
\end{array}
\]

Obviously, the rectangle is a pushout diagram, but the pushout complement we have found is not the only solution as the next figure shows:

\[
\begin{array}{c}
I : \{a_1, b_1\} \\
g_2 : x_1 \mapsto x_5 \\
C_2 : \{[a_5, b_5], d_5, e_5\} \\
\bar{p}_2 : x_5 \mapsto x_3 \\
G : \{[a_3, b_3], c_3, d_3, e_3\}
\end{array}
\]

It easy to see that this diagram is also a pushout diagram with the same $p$ and $\bar{g}$. Since $g_2$ is not an injection, it can not be part of a coproduct construction.\(^2\)

\(^2\)B.K. Rosen [75] calls the first solution (with an injective $g$) an EPS complement because it was introduced in [39] and the second (with an injective $\bar{p}$) the natural complement.
This additional solution essentially makes use of the fact that \( p \) is not injective, and therefore, \( \bar{g} \cdot p = \bar{p} \cdot g \) is not injective. We can choose between \( \bar{p} \) being not injective (first solution) or \( g \) being not injective (second solution). In larger examples, both mappings may be noninjective. We postpone characterizing all solutions in \( \text{Set} \), until we discuss noninjective handles.

If we, however, restrict \( p \) to injections, then \( \bar{p} \cdot g \) is an injection, and therefore, \( g \) is an injection, too, because of Lemma 2.3.3. This results in a more precise version of the previous corollary:

**Corollary 4.2.9 (Injective pushout complements in \( \text{Set} \)):**

In \( \text{Set} \), we have: If \( g \) is an injection, then there exists exactly one pushout complement such that \( g \) is an injection. If in addition, \( p \) is an injection, the pushout complement is uniquely determined.

The reader may ask whether a similar statement on uniqueness can be proved in arbitrary categories. But this is not possible as the category \( \text{Setincl} \) shows:

**Example 4.2.10:**

As we have already seen in Example 4.2.3, the coproduct complement may be ambiguous in \( \text{Setincl} \). Each coproduct complement can be used to construct a pushout complement. The figure shows two possible solutions:

This example shows that even in the case that both \( p \) and \( \bar{g} \) are monomorphisms, the pushout complement is not determined uniquely as long as we allow arbitrary categories.

### 4.3 The General Case

In the categories of interest, monomorphisms have coproduct complements. Therefore, we can apply Lemma 4.2.5 to subdiagrams (2) and (4) of the \( \mathcal{E} - \mathcal{M} \)-decomposition:
The situations we find in subdiagrams (1) and (3) are more complicated, even in the category $\mathcal{S}et$. Subdiagram (1) need not exist, and subdiagram (3) usually is not unambiguous.

The situation we have in subdiagram (1) is similar to that in (2):  

**Corollary 4.3.1:**  
If in the following diagram, $(p_m, \bar{m})$ is a coproduct:

$$
\begin{array}{c}
I \xrightarrow{p_m} B \xrightarrow{\bar{m}} \bar{I} \\
\downarrow g_e \quad = \quad \downarrow \bar{g}_e \quad = \\
C \xrightarrow{p_m} \bar{G}
\end{array}
$$

then $\bar{g}_e \cdot p_m = \bar{p}_m \cdot g_e$ is a pushout if and only if $(\bar{p}_m, \bar{m}')$ is a coproduct.

This corollary is an immediate consequence of Lemma 4.2.4, since we can replace the morphism $g$ in that lemma by $p_m$ and $p$ by $g_e$. As we have already mentioned, the proof is not constructive, since existence of $g_e$ is part of the assumption. In $\mathcal{S}et$, however, we are able to construct the pushout complement if it exists:

**Lemma 4.3.2:**  
If in $\mathcal{S}et$, we have an injective $p_m$ and a surjective $\bar{g}_e$ that satisfies the identification condition, the pushout complement of $\bar{g}_e \cdot p_m$ can be constructed as follows:

(a) Construct $\bar{m}$ as the (unique) coproduct complement of $p_m$.
(b) Construct $\bar{p}_m$ as the coproduct complement of $\bar{m}' := \bar{g}_e \cdot \bar{m}$.
(c) Define $g_e := (\bar{p}_m)^{-1} \cdot \bar{g}_e \cdot p_m$, where $(\bar{p}_m)^{-1}$ is the unique inverse of the restriction $\bar{p}_m : C \to \bar{p}_m[C]$.

Proof:

$\bar{I}$ consists of the elements of $B$ that are not images under $p_m$. Although $\bar{g}_e$ is not an injection, $\bar{m}'$ is since $\bar{g}_e$ satisfies the identification condition (Lemma 4.1.2), and therefore, does not put together elements coming from $I$. This means that we can perform the second step. The last step is possible because $\bar{p}_m$ is injective by construction. Thus, the left-hand part of the diagram is commutative and a pushout diagram by Corollary 4.3.1. The solution is unambiguous because of Lemma 4.2.4 and the fact that in $\mathcal{S}et$, the coproduct complement of $\bar{m}' := \bar{g}_e \cdot \bar{m}$ is unambiguous. Furthermore, $g_e$ is injective because $C$ consists of the elements of $G$ (or $B$, respectively) that do not have a pre-image in $\bar{I}$. i.e., they have a pre-image in $I$. $\square$

**Example 4.3.3:**

We illustrate this lemma by the following example:

$$
\begin{array}{c}
I : \{1, 3, 4\} \xrightarrow{p_m} B : \{1, 3, 4, 5\} \xleftarrow{\bar{m}} \bar{I}' : \{5\} \\
\downarrow \quad \downarrow \quad \downarrow \\
C : \{1, [3, 4]\} \xrightarrow{p_m} G : \{1, [3, 4], 5\}
\end{array}
$$

The morphisms are indicated by the numbers.
In this example, the pushout complement exists and is unambiguous. If, however, the identification condition is not satisfied, we can construct the coproduct complements as before, but it is impossible to find a morphism $g_e$ such that the square commutes (Exercise 4.8.6).

There is still one situation in the $\mathcal{E}-\mathcal{M}$-decomposition that we have not yet considered, namely the pushout complement of two epimorphisms. The so-called natural pushout complement always exists:

**Lemma 4.3.4** (Ehrig/Kreowski [29]):

If in the following diagram

\[
\begin{array}{ccc}
 I & \xrightarrow{p} & B \\
 \downarrow{\bar{g} \cdot p} & & \downarrow{\bar{g}} \\
 G & \xrightarrow{id} & \bar{G}
\end{array}
\]

$p$ and $\bar{g}$ are epimorphisms, then this diagram is a pushout diagram.

**Proof:**

Commutativity is trivial. To show the pushout property, we consider two morphisms $f$ and $g$ with $g \cdot \bar{g} \cdot p = f \cdot p$:

\[
\begin{array}{ccc}
 I & \xrightarrow{p} & B \\
 \downarrow{\bar{g} \cdot p} & & \downarrow{\bar{g}} \\
 G & \xrightarrow{id} & \bar{G} \\
 & \xrightarrow{g} & \downarrow{f} \\
 & & \bar{G}
\end{array}
\]

Then, $g$ itself satisfies the universal property $g = g \cdot id_G$ and $f = g \cdot \bar{g}$, since $p$ is epimorphic. If there were another morphism $h : G \rightarrow H$ with the same property, we would get $h \cdot id_G = g$.

\[\square\]

In general, this natural pushout complement is not the only solution. Let us again consider an example in $\mathbf{Set}$:

**Example 4.3.5:**

We start from the following situation:

\[
\begin{array}{ccc}
 I : \{1, 2, 3, 4, 5, 6\} & \xrightarrow{p} & B : \{[1, 2], [3, 4], 5, 6\} \\
 & \downarrow{\bar{g}} & \\
 G : \{[1, 2], [3, 4], [5, 6]\}
\end{array}
\]

The natural pushout complement is as follows:
The reader can easily verify that the following diagram also is a pushout diagram:

\[ I : \{1, 2, 3, 4, 5, 6\} \rightarrow B : \{[1, 2], [3, 4], 5, 6\} \]
\[ \bar{g} \cdot p \]
\[ C = G : \{[1, 2], [3, 4], [5, 6]\} \rightarrow G : \{[1, 2], [3, 4], [5, 6]\} \]

Therefore, \( g' \) and \( \bar{p}' \) define a pushout complement, too. Comparing these two solutions, we discover that the natural pushout complement accomplishes the task of putting together elements already in the first step \((\bar{g} \cdot p)\), whereas \( g' \) puts together only those elements that must be put together, but are not by the given morphism \( p \).

The solution \( g' \) implements the case with \( p(y_1) \neq p(y_2) \), but \( g(y_1) = g(y_2) \). In this simple example where \( \bar{g} \) puts together only two elements, the remaining solutions can be easily found by applying Corollary 2.5.9: We consider all the pairs \( p(y_1) = p(y_2) \) and modify \( g' \) by putting together some of these combinations.

\[
\begin{array}{ccccccc}
& 1 & 2 & 3 & 4 & 5 & 6 \\
\begin{array}{ccccccc}
g' = g_0 & 1 & 2 & 3 & 4 & [5, 6] & [5, 6] \\
g_1 & [1, 2] & [1, 2] & 3 & 4 & [5, 6] & [5, 6] \\
\bar{g} \cdot p = g_3 & [1, 2] & [1, 2] & [3, 4] & [3, 4] & [5, 6] & [5, 6]
\end{array}
\end{array}
\]

\( g_3 \) is the natural pushout complement.

\( \bar{g} \cdot p \) and \( g' \) can be considered the extremal cases in the class of all possible solutions. Other solutions can be found by splitting the morphism \( \bar{p}_0 \). On the left-hand side of the following figure, you can see another solution to our example. On the right-hand side, we apply this splitting to Example 4.2.8:
In some sense, the natural pushout complement is the *maximal* solution to the problem of finding pushout complements, since we can not find another solution such that its complement object \( C' \) is between the \( C \) of the natural pushout complement and the given \( G \).

On the contrary, if we have a pushout complement with a \( \bar{p} \) that is not monomorphic, we can construct “greater” solutions by shifting epimorphic parts from the bottom arrow to the left-hand arrow:

**Lemma 4.3.6** (Ambiguous pushout complements):

Let \( \bar{g} \cdot p = \bar{p} \cdot g \) be a pushout diagram. If there is a factorization \( \bar{p} = \beta \cdot \alpha \) with an epimorphic \( \alpha \) and an arbitrary \( \beta \), then \( \bar{g} \cdot p = \beta \cdot (\alpha \cdot g) \) is a pushout diagram, too.

Proof:

We consider the following diagram and assume \( k \cdot p = h \cdot (\alpha \cdot g) \). We can rewrite this as \( k \cdot (p \cdot \text{id}) = (h \cdot \alpha) \cdot g \). Since the outer diagram is a pushout, we get a unique morphism \( q : G \rightarrow H \) with \( k = q \cdot \bar{g} \) and \( h \cdot \alpha = q \cdot (\beta \cdot \alpha) = (q \cdot \beta) \cdot \alpha \) and with \( \alpha \) assumed to be epimorphic: \( h = q \cdot \beta \).

Please note that this lemma is not restricted to pushout complements of two epimorphisms. It is applicable to each pushout complement. In an \( \mathcal{E} - \mathcal{M} \)-factorizable category, we can split \( \bar{p} \) into an epimorphism \( \beta \) and a monomorphism \( \alpha \), unambiguously, and we get the following corollary:

**Corollary 4.3.7:**

If in an \( \mathcal{E} - \mathcal{M} \)-factorizable category, \( \bar{g} \cdot p \) has a pushout complement, then a pushout complement \( \bar{g} \cdot p = \bar{p} \cdot g \) with a monomorphic \( \bar{p} \) exists.

If we know a *minimal* pushout complement, the lemma allows us to construct other pushout complements. Furthermore, a minimal pushout complement plays an important role in derivation steps: It gives us as much freedom as possible in completing the
right-hand side, since $g$ does not put together elements that are already put together along $p$. In general, however, we do not have a unique minimal pushout complement even in the category $\text{Set}$:

**Example 4.3.8:** We consider the following situation:

$$
\begin{align*}
\{1, 2, 3, 4, 5, 6\} \xrightarrow{p} \{1, [2, 3], [4, 5], 6\} \\
\xrightarrow{g} \{[1, 2, 3, 4, 5, 6]\}
\end{align*}
$$

Both $\{[1, 2], [3, 4], [5, 6]\}$ and $\{[1, 4], [2, 6], [3, 5]\}$ are minimal solutions. It is left to the reader to find further solutions.

### 4.4 Constructing Pushout Complements in $\text{Set}$

Now, we have studied all the subproblems, and we can summarize the construction in the category of sets:

\begin{itemize}
  
  (a) Construct $\bar{m}$ as the coproduct complement of $p_m$.
  
  (b) If the identification condition is satisfied, $\bar{m}'$ is injective, and $p'_m$ can be constructed as the coproduct complement of $\bar{m}'$.
  
  (c) Construct $\bar{g}_m'$ as the coproduct complement of $\bar{g}_m$.
  
  (d) Construct the coproduct $(g'_m, g''_m)$. The morphism $\bar{p}_m$ is defined by the universal property of this coproduct.
  
  (e) Choose a pushout complement of $g'_e \cdot p_e$. At least, the natural pushout complement exists. Of course, the result of step (g) depends on this choice.
  
  (f) Construct the coproduct complement $\bar{g}_m''$ of $g'_m$.
\end{itemize}
(g) Construct the coproduct \((g_m, g'_m)\). The morphism \(\bar{p}_e\) is given by the universal property of \((g_m, g'_m)\).

The solution exists if \(\bar{g}_e\) satisfies the identification condition. The set of all possible solutions can be characterized by considering all the pushout complements of \(g'_e \cdot p_e\). Please note that the noninjective alternatives to \(g_m\) (Corollary 4.2.7) are excluded by the decomposition.

The following example illustrates the computational aspects of finding pushout complements in \(\text{Set}\). We present it in full details because it is the basis of constructing pushout complements in \(\text{Graph}\).

**Example 4.4.1:**

We start from the following situation:

\[
\begin{align*}
\{1, 2, 3, 4\} & \rightarrow \{[1, 2], 3, 4, 5\} \\
& \rightarrow \{[1, 2], [3, 4], 5, 6\}
\end{align*}
\]

and decompose both morphisms into surjective and injective parts:

\[
\begin{align*}
\{1, 2, 3, 4\} & \rightarrow \{[1, 2], 3, 4\} \rightarrow \{[1, 2], 3, 4, 5\} \\
& \rightarrow \{[1, 2], [3, 4], 5\} \\
& \rightarrow \{[1, 2], [3, 4], 5, 6\}
\end{align*}
\]

The identification condition is satisfied. Therefore, we can complete the subdiagram (1) using the technique presented in Example 4.3.3:

\[
\begin{align*}
\{1, 2, 3, 4\} & \rightarrow \{[1, 2], 3, 4\} \rightarrow \{[1, 2], 3, 4, 5\} \rightarrow \{5\} \\
& \rightarrow \{[1, 2], [3, 4], 5\} \\
& \rightarrow \{[1, 2], [3, 4], 5, 6\}
\end{align*}
\]

Next, we complete subdiagrams (2) and (3). To find subdiagram (2), we apply Lemma 4.2.5. In the case of subdiagram (3), we choose the natural pushout diagram (Lemma 4.3.4):
Finally, we again apply Lemma 4.2.5 to construct subdiagram (4), and we get the first solution shown on top of the next page. Possible ambiguities are restricted to subdiagram (3), but the choice of a solution to (3) influences the final result. In our small example, we have only one alternative to subdiagram (3) resulting in the diagram also given on the next page.

If in a category, the coproduct complement is unique if it exists, the general construction of pushout complements (page 123) can yield ambiguous solutions only in subdiagram (3). Due to Lemma 4.3.4, this subdiagram has at least one solution, namely the natural pushout complement of epimorphisms. In this case, $p_e'$ is an isomorphism, and since pushouts preserve isomorphisms, the same holds for $\bar{p}_e$, i.e., $\bar{p} = \bar{p}_m \cdot \bar{p}_e = \bar{p}_m$. This allows us to generalize the notion of the natural pushout complement:

**Lemma 4.4.2** (Natural pushout complement, [75]):

If in an $\mathcal{E}–\mathcal{M}$-factorizable category, monomorphisms have a unique coproduct complement, each pair of composable morphisms $g \cdot p$ has at least one pushout complement $\bar{g} \cdot p = \bar{p} \cdot g$ such that $\bar{p}$ is a monomorphism. We call these pushout complements the natural pushout complements.
Second solution to Example 4.4.1.

In the category \( \text{Set}_{\text{incl}} \), each morphism is both a monomorphism and an epimorphism (Example 2.3.9). Therefore, each pushout complement is a natural pushout complement. This means that in general, the natural pushout complement may be ambiguous. In the categories \( \text{Set} \) and \( \text{Graph} \), however, the natural pushout complement is unambiguous. This can be easily seen along the lines of Example 4.3.5. If you do not choose the identity in subdiagram (3), it is necessary to put some elements together.

**Corollary 4.4.3** (Natural pushout complement in \( \text{Set} \)):

If in \( \text{Set} \) or in \( \text{Graph} \), a pair of composable morphisms \( \bar{g} \cdot p \) has pushout complements \( \bar{g} \cdot p = \bar{p} \cdot g \), then there is a unique pushout complement with \( \bar{p} \) being monomorphic.

### 4.5 The Situation in \( \text{Graph} \)

We now resume discussing the problem we are mainly interested in, namely the pushout complement in \( \text{Graph} \). Of course, we can expect that a solution exists only if we can find solutions for both the nodes and the edges, since a pushout in \( \text{Graph} \) implies that the subdiagrams for nodes and edges are pushouts in \( \text{Set} \) (Exercise 3.8.1). What about the converse?

**Lemma 4.5.1:**

If we have a commutative diagram \( \bar{g} \cdot p = \bar{p} \cdot g \) in \( \text{Graph} \) such that both the subdiagram for edges and the subdiagram for nodes are pushout diagrams in \( \text{Set} \), the whole diagram is a pushout diagram in \( \text{Graph} \):

\[
\begin{array}{ccc}
I & p & B \\
\text{g} & = & \text{g} \\
C & p & G \\
\end{array} \quad \begin{array}{ccc}
I_E & p_E & B_E \\
\text{g}_E & = & \text{g}_E \\
C_E & p_E & G_E \\
\end{array} \quad \begin{array}{ccc}
I_V & p_V & B_V \\
\text{g}_V & = & \text{g}_V \\
C_V & p_V & G_V \\
\end{array}
\]

Proof:
We start with two graph morphisms $\bar{g}'$ and $\bar{p}'$ satisfying $\bar{g}' \cdot p = \bar{p}' \cdot g$. From this, we get $\bar{g}'_E \cdot p_E = \bar{p}'_E \cdot g_E$, and since the subdiagram for the edges is a pushout, there exists a unique $h_E$ with $\bar{p}'_E = h_E \cdot \bar{p}_E$ and $\bar{g}'_E = h_E \cdot \bar{g}_E$. The same holds for the nodes.

Now, we consider the right-hand diagram connecting the nodes of $H$ to the pushout diagram for the edges by the source node function $s_H : H_E \to H_V$. From this diagram, we get a unique $k$ such that $s_H \cdot h_E \cdot \bar{g}_E = h_V \cdot s_G \cdot \bar{g}_E = k \cdot \bar{g}_E$ and $s_H \cdot h_E \cdot \bar{p}_E = h_V \cdot s_G \cdot \bar{p}_E = k \cdot \bar{p}_E$. Since both $s_H \cdot h_E$ and $h_V \cdot s_G$ satisfy this property, they must be equal up to isomorphism, i.e., $(k_E, k_V)$ is a graph morphism. □

Therefore, we can reduce our problem to the following construction:

**Construction 4.5.2** (Componentwise construction of pushout complements):

We can solve the problem of finding a pushout complement in $\text{Graph}$ by constructing pushout complements for edges and nodes, separately. If we can ensure that the resulting morphisms $(g_E, g_V)$ and $(\bar{p}_E, \bar{p}_V)$ are graph morphisms, we have a commutative diagram in $\text{Graph}$, and the lemma tells us that this is a pushout diagram in $\text{Graph}$.

In general, the resulting morphisms do not constitute graph morphisms. The remaining part of this section is dedicated to elaborate some restrictions ensuring this property.

**Example 4.5.3**:

We consider an example reflecting the simplest case, i.e., a monomorphic production and a monomorphic handle:

In this example, we can construct unique pushout complements for both nodes and edges according to Corollary 4.2.7. Let us consider what happens in $\text{Graph}$. 
For the moment, we connect edges to nodes by intuition as shown in the figure on top of the next page. As the example shows, constructing the coproduct complement of a graph morphism component by component does not result in a graph. There are some edges in $G$ that are not images under $\bar{g}$ and therefore, the corresponding edges must occur in $C'$, whereas the source (or target) nodes of them do not. In the figure, we have indicated these missing nodes by bullets. Nevertheless, the pushout complement $C$ is a graph.

Before looking for a criterion ensuring that the pushout complements we construct separately for nodes and edges constitute a graph, we consider another example showing what may happen.

**Example 4.5.4:**

We slightly modify the previous example by replacing the edge from node 5 to node 3 by an edge from 5 to node 4. The construction is shown below. It is easy to verify that the diagrams $\bar{g}_V \cdot p_V = \bar{p}_V \cdot g_V$ for the nodes and $\bar{g}_E \cdot p_E = \bar{p}_E \cdot g_E$ for the edges are pushout diagrams. But in $C_V$, it is impossible to find a target node for the edge that starts at node 5. You may try to use any of the nodes 1, 2, 3, or 5: Whichever you choose, $\bar{p}$ is not a graph morphism. (Please note that the construction of the pushout complement of $\bar{g}_V$ and $p_V$ results in $\bar{p}_V(i) = i$.)

If we assume monomorphisms in Graph, Corollary 4.2.7 says that we can construct the pushout complements for nodes and edges. But Example 4.5.4 shows that this is not sufficient to construct the pushout complement in Graph. There is a simple reason for this problem: The edge from 5 to 4 in $G$ has no pre-image in $B$. This means that it must come from $C'_E$, whereas its target node 4 comes from $B_V$ and can not be in $C'_V$,
Failed attempt to solve Example 4.5.4

and therefore, it can not be in $C_V$. We have the same situation considering nodes 2 and 3 and the edge between them. But in this case, the nodes are interface nodes and therefore, they remain in $C_V$!

This observation leads to the following criterion:

**Lemma 4.5.5:**

Let $p$ and $\bar{g}$ be two monomorphisms in $\mathcal{G}raph$. Then, there exist unambiguous pushout diagrams $\bar{g}_V \cdot p_V = \bar{p}_V \cdot g_V$ and $\bar{g}_E \cdot p_E = \bar{p}_E \cdot g_E$ for nodes and edges, respectively, that can be completed to a pushout diagram in $\mathcal{G}raph$ by defining $s_C$ and $t_C$ suitably if and only if the following dangling condition holds:

$$s_G\bar{g}_E[C'_E] \subseteq \bar{g}_V'[C'_V] \cup \bar{g}_V p_V[I_V]$$

$$t_G\bar{g}_E[C'_E] \subseteq \bar{g}_V'[C'_V] \cup \bar{g}_V p_V[I_V]$$

where $\bar{g}_E'$ and $\bar{g}_V'$ are the mappings introduced according to Lemma 4.2.5:

Proof:

The diagrams show the componentwise construction. By Corollary 4.2.7, the pushout
complement of edge mappings as well as the pushout complement of node mappings are unambiguously determined.

First, we show that the dangling condition follows from the pushout property in $\mathbf{Graph}$. If $\bar{p} \cdot g = \bar{g} \cdot p$ is a pushout in $\mathbf{Graph}$, $\bar{p}$ is a graph morphism, i.e., $s_G \cdot \bar{p} = \bar{p}_V \cdot s_C$. By construction, we have $s_G \cdot \bar{g}_E = s_G \cdot \bar{p} \cdot g_E = \bar{p}_V \cdot s_C \cdot g_E$, and we must show that the codomain of this mapping satisfies the dangling condition. Since $C_V$ is the coproduct object of $I_V$ and $C'_V$, i.e., $C_V = g_V[I_V] \cup g'_V[C'_V]$, we get $s_G \bar{g}_E : C'_E \to C_V = g_V[I_V] \cup g'_V[C'_V]$, and therefore, $s_G \bar{g}_E[C'_E] \subseteq \bar{p}_V g'_V[C'_V] \cup \bar{p}_V g_v[I_V] = \bar{g}'_V[C'_V] \cup \bar{g}_V p_V[I_V]$. The argument for $t_G$ is analogous.

To prove the converse, we have to construct $s_C$. The idea is to use the universal property of the coproduct $(g_E, \bar{g}_E)$. For this, we need a morphism from $C'_E$ to $C_V$, which we call $\gamma$:

After following $\bar{g}_E$ and $s_G$, we reach elements in $G_V$ which may come from $B_V$ or from $C'_V$. The coproduct property of $(\bar{g}_E, \bar{g}_V)$ ensures that this distinction is unambiguous. In addition, the dangling condition ensures that elements in $s_G \bar{g}_E$ that come from $B_V$ are interface nodes, i.e., they are in $\bar{g}_V p_V[I_V]$. In this case, we choose an inverse of $\bar{g}_V \cdot p_V$, followed by $\bar{g}_V$. In the other case, we choose the inverse of $\bar{g}'_V$, followed by $g'_V$. In this way, we get a morphism $\gamma$ connecting $C'_E$ to $C_V$. We can formalize this idea in the following way:

(a) We restrict the codomain of $\bar{g}_V \cdot p_V : I_V \to G_V$ such that it becomes surjective:

$r : I_V \to \bar{g}_V p_V[I_V]$. $r$ is a retraction in $\mathbf{Set}$, i.e., there exists a morphism $\bar{r} : \bar{g}_V p_V[I_V] \to I_V$ such that $\bar{r} \circ \bar{r} = \text{id}_{\bar{g}_V p_V[I_V]}$, and we have $\bar{g}_V p_V \bar{r} = \text{id}_{\bar{g}_V p_V[I_V]}$.

(b) Analogously, we restrict $\bar{g}'_V : C'_V \to G_V$ considering $q : C'_V \to \bar{g}'_V[C'_V]$ and an inverse $\bar{q} : \bar{g}'_V[C'_V] \to C'_V$ with $\bar{q} \circ \bar{q} = \text{id}_{\bar{g}'_V[C'_V]}$. Here, we have $\bar{g}'_V \bar{q} = \text{id}_{\bar{g}'_V[C'_V]}$.

(c) Thus, we get a morphism $\beta : \bar{g}_V p_V[I_V] \cup \bar{g}'_V[C'_V] \to C_V$ by:

$$\beta(v) := \begin{cases} g'_V(\bar{q}(v)) & \text{if } v \in \bar{g}'_V[C'_V] \\ g_V(\bar{r}(v)) & \text{if } v \in \bar{g}_V p_V[I_V] \end{cases}$$
(d) Finally, we define $\gamma := \beta \cdot s_G \cdot \bar{g}_E$.

It is easy to verify that $\bar{p}_V \cdot \beta$ is the identity on $\bar{g}_V p_V [I_V] \cup \bar{g}_V [C'_V]$ and $\bar{p}_V \cdot \gamma = s_G \cdot \bar{g}_E$.

Since $(g'_E, g_E)$ is a coproduct, we can factorize $\gamma$ and $g_V \cdot s_I$ via a common $s_C$ such that $\gamma = s_C g'_E$ and $g_V \cdot s_I = s_C g_E$, i.e., $g$ is a graph morphism. Furthermore, we have to show that $\bar{p}$ is a graph morphism, too, i.e., $s_G \bar{p}_E = \bar{p}_V s_C$. This can be done by taking advantage of the coproduct property of $(g'_E, g_E)$ again: We consider the morphisms $s_G \bar{p}_E g_E$ and $\bar{p}_V s_C g'_E$ connecting $I_E$ and $C'_E$, respectively, to $G_V$. By the coproduct property of $C_E$, we get the existence of a unique $h : C_E \rightarrow G_V$ such that $s_G \bar{p}_E g_E = h g_E \land \bar{p}_V s_C g'_E = h g'_E$. Both $h = s_G \bar{p}_E$ and $h = \bar{p}_V s_C$ satisfy these conditions, since we have $s_G \bar{p}_E g_E = s_G \bar{g}_E = \bar{p}_V \gamma = \bar{p}_V s_C g'_E$ and $\bar{p}_V s_C g_E = \bar{p}_V g_V s_I = \bar{g}_V p_V [I_V] \cup \bar{g}_V [C'_V]$.

Now, we know that the constructed diagram $\bar{g} \cdot p = \bar{p} \cdot g$ is a commutative diagram in $\mathcal{Graph}$ and it is a pushout diagram in $\mathcal{Graph}$ because of Lemma 4.5.1.

In Lemma 4.5.5, we have assumed a monomorphic graph morphism $p$ for reasons of simplicity. If we abandon this restriction, Lemma 4.2.5 still is applicable, but the solution is no longer unambiguous. Decomposition of pushouts (Corollary 4.1.9) allows us to restrict the discussion to epimorphic graph morphisms $p$.

**Lemma 4.5.6:**

If $p$ is an epimorphism and $\bar{g}$ is a monomorphism in $\mathcal{Graph}$, then there exist pushout diagrams $\bar{g}_V \cdot p_V = \bar{p}_V \cdot g_V$ and $\bar{g}_E \cdot p_E = \bar{p}_E \cdot g_E$ for nodes and edges, respectively, that can be completed to pushout diagrams in $\mathcal{Graph}$ by defining $s_C$ and $t_C$ if and only if the dangling condition of Lemma 4.5.5 holds.

**Proof:**

As in the previous proof, we can construct pushout complements of the node mappings and of the edge mappings, separately. This construction is unambiguous, although it does not yield all possible solutions as we have seen in Example 4.2.8. The first part of the proof did not use the property of $p$ to be monomorphic. Proving the converse, $\bar{r}$ is no longer unambiguous. For each $\bar{r}$, we get a morphism $\gamma_{\bar{r}}$ depending on that $\bar{r}$. As you may easily check, each $\gamma_{\bar{r}}$ satisfies the arguments of the previous proof.

**Example 4.5.7:**

Let us consider an example with a noninjective $p_V$ as shown in the figure. Here, there are two ways to define $r$:

\[
\begin{align*}
  r_1(1') & = 1 & r_2(1') & = 1 \\
  r_1([2', 3']) & = 2 & r_2([2', 3']) & = 3
\end{align*}
\]

This leads to two different $\gamma_1$ and $\gamma_2$, and therefore, to two different pushout complements $C_1$ and $C_2$:

\[
C_1 = \begin{array}{ccc}
  1 & \rightarrow & 2 \\
  \downarrow & & \downarrow \\
  3 & \rightarrow & 4 \\
\end{array}
\quad C_2 = \begin{array}{ccc}
  1 & \rightarrow & 2 \\
  \downarrow & & \downarrow \\
  3 & \rightarrow & 4 \\
\end{array}
\quad C_3 = \begin{array}{ccc}
  1 & \rightarrow & [2, 3] \\
  \downarrow & & \downarrow \\
  4 & \rightarrow & 4 \\
\end{array}
\]

\[3\bar{q} \text{ is unambiguous since } \bar{g}_V \text{ is injective by construction.}\]
Both solutions result in graph morphisms. Obviously, we can not find the third solution $C_3$ by this way, since the coproduct construction can not put together two nodes.

In this example, we have considered a noninjective $p_V$. Exercise 4.8.8 considers the case of a noninjective $p_E$.

Lemmata 4.5.5 and 4.5.6 treat all cases with monomorphic handles. We can summarize these results as follows:

**Theorem 4.5.8** (Dangling condition, Ehrig/Pfender/Schneider, 1973 [39]):

Let $p$ be an arbitrary morphism in $\text{Graph}$ and $\bar{g}$ a graph monomorphism. Then, there exist pushout diagrams $\bar{g} \cdot p = \bar{p} \cdot g$ if and only if the following condition holds:

\[
\begin{align*}
  s_G \bar{g}_E'[C'_E] &\subseteq \bar{g}'_V[C'_V] \cup \bar{g}_V p_V[I_V] \\
  t_G \bar{g}_E'[C'_E] &\subseteq \bar{g}'_V[C'_V] \cup \bar{g}_V p_V[I_V]
\end{align*}
\]

with $\bar{g}_E'$ and $\bar{g}_V'$ as in Lemma 4.2.5. In the case of a monomorphic $p$, the solution is unique.

Now, we can explain why we have not made the transition node an interface node in the Petri net example (Example 3.6.1). The dangling condition states that an edge of $G$ that is not in $B$ must be connected to nodes that are not in $B$, too, or that are interface nodes. Therefore, each edge connected to the transition must be in $B$. (Otherwise, the pushout complement does not exist.) This ensures that the production takes into consideration all input places and all output places. On the other hand, it is impossible that nodes not mentioned in the production are connected to the transition.

Now, we switch over to studying epimorphic handles. If $p$ is epimorphic, too, Lemma 4.3.4 ensures the existence of the natural pushout complement. Otherwise, we can
Example 4.5.9

We consider a monomorphic $p$ and an epimorphic $\bar{g}$ as shown on top of the next page. It is easy to see that the left-hand part of this diagram is a pushout. The edges $1 \to 2$ of $B$ and $1 \to [2,3]$ of $C$ are thrown together because of the common pre-image in $I$. The second edge $1 \to [2,3]$ in $G$ is the image of the edge $1 \to 3$ in $B$.

In this small example, it is obvious how to connect the edge of $C$ to its nodes. This straightforward way can be always applied:

**Lemma 4.5.10:**

If we have a monomorphic graph morphism $p$ and an epimorphic graph morphism $\bar{g}$ such that both $\bar{g}_E$ and $\bar{g}_V$ satisfy the identification condition (Lemma 4.1.2), then we can unambiguously combine the pushout complements of $\bar{g}_E \cdot p_E$ and $\bar{g}_V \cdot p_V$ to yield a pushout complement in $\mathcal{G}raph$ by defining $s_C$ and $t_C$.

Proof:

We construct the pushout complements for nodes and edges, separately, according to Lemma 4.3.3 as shown in the figure. Whereas $s_I$, $s_B$, and $s_G$ are given, we
need a suitable morphism $s_C$ to completely connect the subdiagrams. We define $s_C := g_V \cdot s_I \cdot (g_E)^{-1}$ with a coretraction $(g_E)^{-1}$ of $g_E$. Since the construction of Lemma 4.3.3 yields an epimorphic $g_E$, this coretraction exists, but need not be unambiguous. Nevertheless, $s_C$ does not depend on the choice of the coretraction. To prove this, we consider two edges $e, e' \in I_E$ with $g_E(e) = g_E(e')$ and therefore, $\bar{p}_E(g_E(e)) = \bar{p}_E(g_E(e'))$, which can be rewritten as $\bar{g}_E(p_E(e)) = \bar{g}_E(p_E(e'))$. Since $\bar{p}_V \cdot g_V \cdot s_I = \bar{g}_V \cdot p_V \cdot s_I = \bar{g}_V \cdot s_B \cdot p_E$, we get $\bar{p}_V(g_V(s_I(e))) = s_G(\bar{g}_E(p_E(e))) = s_G(\bar{g}_E(p_E(e'))) = \bar{p}_V(g_V(s_K(e'))) = g_V(s_K(e'))$ because $\bar{p}_V$ is monomorphic.

This $s_C$ and the analogously defined $t_C$ make $g$ and $\bar{p}$ graph morphisms. In the case of $\bar{p}$, we have $\bar{p}_V \cdot s_C = p_V \cdot g_V \cdot s_I \cdot (g_E)^{-1} = \bar{g}_V \cdot p_V \cdot s_I \cdot (g_E)^{-1} = g_V \cdot s_B \cdot p_E \cdot (g_E)^{-1} = s_G \cdot \bar{g}_E \cdot p_E \cdot (g_E)^{-1} = s_G \cdot \bar{p}_E \cdot g_E \cdot (g_E)^{-1} = s_G \cdot \bar{p}_E$. On the other hand, we have $s_C \cdot g_E = g_V \cdot s_I \cdot (g_E)^{-1} \cdot g_E$. In general, $(g_E)^{-1} \cdot g_E$ is not the identity if $g_E$ is a retraction, but in our case, the ambiguity is removed by $g_V \cdot s_I$ as we have already seen. Thus, we get $s_C \cdot g_E = g_V \cdot s_I$.

\begin{example}[Cont’d]
We slightly modify the example such that $(g_E)^{-1}$ is no longer unambiguous:

In this case, $\bar{g}$ puts together the edges $1 \to 2$ and $1 \to 3$ of $B$. The identification condition is satisfied because both edges are interface edges. The edge $1 \to [2,3]$ in graph $C$ has two pre-images in $I$. Therefore, we have two possible coretractions of $g_E$, but the target nodes are put together, too!

In Graph, we have not yet considered the case of two epimorphisms $p$ and $\bar{g}$. Trivially, the natural pushout complement exists. In Section 4.3, we have seen that there are other pushout complements in Set. If we combine an arbitrary solution for nodes with an arbitrary solution for edges, this does not necessarily result in a pushout complement in Graph:

\begin{example}
We consider an interface graph with four nodes and three edges. Graph morphism $p$ puts together the nodes 1 and 2 and the edges 5 and 6, respectively, whereas $\bar{g}$ is the identity as shown in the figure on top of the next page. Constructing the pushout complement for the nodes and the edges separately in Set, we may put the elements together by the morphisms $g_E$ and $g_V$ or we let them distinct. If we choose a morphism $g_E$ putting together the edges 5 and 6, but a morphism $g_V$
not putting together the nodes 1 and 2, \( g \) is not a graph morphism, and therefore, we do not get a pushout complement in \( \mathcal{G}raph \), since \( s_C(g_E(6)) \neq s_K(6) \).

**Lemma 4.5.12** (Compatibility condition, Ehrig/Kreowski [29]):
Let be \( p \) and \( \bar{g} \) epimorphisms in \( \mathcal{G}raph \) and let be \( g_E, \bar{g}_E \) and \( g_V, \bar{g}_V \) arbitrary pushout complements of \( p_E, \bar{p}_E \) and \( p_V, \bar{p}_V \), respectively. Then, we can define \( s_C \) and \( t_C \) such that \((g_E, g_V)\) and \((\bar{p}_E, \bar{p}_V)\) make up a pushout complement in \( \mathcal{G}raph \) if and only if \( g_E \) and \( g_V \) satisfy:

\[
(\forall e, e' \in E_E)(g_E(e) = g_E(e') \Rightarrow g_V(s_I(e)) = g_V(s_I(e')) \\
\wedge g_V(t_I(e)) = g_V(t_I(e')))
\]

This condition is not surprising: it simply means that if two edges are put together, their source and target nodes must be put together, too. The proof is analogous to the proof of Lemma 4.5.10. The only difference is that we can not derive the equality \( \bar{p}_V(g_V(s_I(e))) = \bar{p}_V(g_V(s_I(e'))) \) from the construction, but we get it from the compatibility condition.

Now, we can give the details of Construction 4.5.2:

**Construction 4.5.13** (Construction of pushout complements in \( \mathcal{G}raph \)):
We can construct pushout complements in \( \mathcal{G}raph \) as follows:

(a) Construct pushout complements for edges and nodes, separately. This is possible if both satisfy the identification condition (Lemma 4.1.2).
(b) If \( \bar{g} \) is a monomorphism, the dangling condition (Theorem 4.5.8) ensures the existence of pushout complements in \( \mathcal{G}raph \).
(c) If \( \bar{g} \) is an epimorphism and \( p \) is a monomorphism, we can complete the diagram in \( \mathcal{G}raph \) without problems (Lemma 4.5.10).
(d) If both morphisms are epimorphic, the natural pushout complement exists in each case. Further solutions exist as long as the combinations of the separate solutions for edges and nodes satisfy the compatibility condition of Lemma 4.5.12.  

\[\text{The natural pushout complement trivially satisfies the compatibility condition.}\]
This completely answers the question of how to find pushout complements in \( \text{Graph} \).

### 4.6 Labeled Graphs and Hypergraphs

Now, we switch over to labeled graphs and label preserving graph morphisms. We have to ask for a suitable labeling of \( C \).

**Theorem 4.6.1** (Pushout complement in \( \mathcal{L}\text{graph} \)):

The construction of pushout complements in \( \text{Graph} \) induces an unambiguous construction of pushout complements in \( \mathcal{L}\text{graph} \).

**Proof:**

We refer to the diagram used in proving Theorem 3.3.1:

But now, \( l_{VG} \) is given and we have to find suitable morphisms \( l_{VC} \) and \( l_{EC} \) that make \( \bar{p} \) and \( g \) morphisms in \( \mathcal{L}\text{graph} \). Trivially, \( \bar{p} \) is an \( \mathcal{L}\text{graph} \)-morphism, if we define \( l_{VC} := l_{VG} \cdot \bar{p} \) and \( l_{EC} := l_{EG} \cdot \bar{p} \). This definition also makes \( g \) a morphism in \( \mathcal{L}\text{graph} \): \( l_{VC} \cdot g_V = l_{VG} \cdot \bar{p} \cdot g_V = l_{VG} \cdot \bar{g} \cdot p_V = l_{VB} \cdot p_V = l_{V1} \).

Therefore, \( \mathcal{L}\text{graph} \) does not add any new aspects to the construction we have already discussed: We construct one or more pushout complements of the underlying graph morphisms and add the labels of the corresponding nodes and edges in \( G \). An example is given as Exercise 4.8.12.

Now, we consider hypergraphs:

**Lemma 4.6.2:**

If we have a commutative diagram \( \bar{g} \cdot p = \bar{p} \cdot g \) in \( \mathcal{H}\text{graph} \) such that both \( \bar{g} \cdot p = \bar{p} \cdot g \) and \( \bar{g} \cdot p = \bar{p} \cdot g \) are pushout diagrams in \( \text{Set} \), the diagram \( \bar{g} \cdot p = \bar{p} \cdot g \) is a pushout diagram in \( \mathcal{H}\text{graph} \).
The proof is analogous to that of Lemma 4.5.1. We have only to replace $H_V$ by $H_Y$.

Therefore, we can use the idea of how to construct pushout complements in $\mathcal{G}raph$ to construct pushout complements in $\mathcal{H}ypergraph$, too. The identification condition does not refer to the graph structure; it is to be checked for nodes and edges, separately. The dangling condition (Theorem 4.5.8) and the compatibility condition (Lemma 4.5.12), however, are essential to make the result a graph. Their intuitive interpretations help us in adapting them to hypergraphs:

**Lemma 4.6.3** (Dangling condition in $\mathcal{H}ypergraph$):

Let $p$ be an arbitrary morphism in $\mathcal{H}ypergraph$ and $g$ a hypergraph monomorphism. Then, there exist pushout diagrams $\bar{g} \cdot p = \bar{p} \cdot g$ if and only if the following condition holds:

$$s_G\bar{g}_E'[C'_E] \subseteq (\bar{g}'_V[C'_V] \cup \bar{g}_V \cdot p_V[I_V])^*$$

$$t_G\bar{g}_E'[C'_E] \subseteq (\bar{g}'_V[C'_V] \cup \bar{g}_V \cdot p_V[I_V])^*$$

with $\bar{g}'_E$ and $\bar{g}'_V$ being the coproduct complements of $\bar{g}_E$ and $\bar{g}_V$, respectively. In the case of a monomorphic $p$, the solution is unique.

Proving the dangling condition for usual graphs (Lemma 4.5.5 and Theorem 4.5.8), we have taken advantage of the fact that $s_G \cdot \bar{g}_E$ maps an edge of $C'_E$ either on a node in $\bar{g}'_V[C'_V]$ or on a node in $\bar{g}_V \cdot p_V[I_V]$. In the case of hypergraphs, however, $s_G(\bar{g}_E(e))$ is a sequence that may consist of nodes in both subsets. Please note that in the formula, the *-operator is applied to the union. We illustrate this situation by slightly modifying Example 4.5.3:

**Example 4.6.4**:

We interpret the edges in $I$ and in $B$ as hyperedges with one source node and with one target node. Of course, this also holds for their images in $G$. We replace the edges from 2 to 3 and from 5 to 3 by a single hyperedge $e$ with the source nodes $s_1(3) = 2$ and $s_2(e) = 5$ and the target node $t_1(e) = 3$. We get the situation shown on top of the next page. Hyperedge $e$ is the edge of interest\(^5\). Its first source node and its target node are interface nodes whereas its second source node is not. Nevertheless, the dangling condition is satisfied.

Proof of the lemma:

We can modify the proof of Lemma 4.5.5 such that it is also applicable to $\mathcal{H}ypergraph$. In the first part showing that the dangling condition follows from the pushout property, we have $s_G \cdot \bar{g}_E = \bar{p}_V \cdot s_C \cdot \bar{g}_E'$ and $s_G \cdot \bar{g}_E'[C'_E] \subseteq (g_V[I_V] \cup g'_V[C'_V])^*$. From this, we get $\bar{p}_V \cdot s_C \cdot \bar{g}_E'[C'_E] \subseteq (\bar{p}_V \cdot g_V[I_V] \cup \bar{p}_V \cdot \bar{g}'_V[C'_V])^*$ considering the sequences component by component. This completes the first part of the proof because of $\bar{p}_V \cdot g_V = \bar{g}_V \cdot p_V$ and $\bar{p}_V \cdot \bar{g}_V' = \bar{g}'_V$.

To prove the converse, we define $\bar{r}$, $\bar{q}$, and $\beta$ as in the case of graphs, but $\gamma := \beta^* \cdot s_G \cdot \bar{g}_E'$. As before, $\bar{p}_V \cdot \beta$ is the identity on $\bar{g}_V \cdot p_V[I_V] \cup \bar{g}_V'[C'_V]$ and therefore, $\bar{p}_V \cdot \beta^*$ the identity on $(\bar{g}_V \cdot p_V[I_V] \cup \bar{g}_V'[C'_V])^*$. Thus, we have $\bar{p}_V \cdot \gamma = \bar{p}_V \cdot \beta^* \cdot s_G \cdot \bar{g}_E = s_G \cdot \bar{g}_E$. The rest of the proof is analogous to that for graphs, i.e., the coproduct property of $(g'_E, g_{CE})$ yields a unique $s_C$ with $\gamma = s_C \cdot g'_E$ and $g'_V \cdot s_I = s_C \cdot g_E$.

\(^5\)For reasons of clearness, we have omitted the other edge identifiers.
Solution to Example 4.6.4

Of course, labeled hypergraphs with label preserving morphisms do not add any new problems.

4.7 Structurally Labeled Graphs

Label preserving morphisms have not added any difficulties to constructing derivation steps effectively. We get very a different situation if we consider a structured alphabet. Of course, we can use the decomposition theorem (Theorem 4.1.8) in the case of struc-
turally labeled graphs, too. The idea of applying this theorem is to consider substeps that take advantage of the special properties of monomorphisms or epimorphisms. Unfortunately, the coproduct complement, which is used to construct the pushout complements in Lemma 4.2.5 (injective handle) and in Lemma 4.3.2 (injective production), need not be unique in the case of a structured alphabet as we have shown for \texttt{Setincl} (Example 4.2.3). Although the underlying graph can be constructed unambiguously, its labeling may be ambiguous. Parisi-Presicce, who studied structured alphabets for the first time [67], has restricted discussion to injective productions and has used the minimal pushout complement\textsuperscript{6} to define derivability. This restriction is not necessary. Considering the left-hand side of the derivation step, we can admit all pushout complements as context graphs. In the second step, we can exclude those complements that do not allow to complete the right-hand side in a suitable manner.

Paris-Presseicce \textit{et al.} have already observed that we can restrict changing labels to the “epimorphic part” of the decomposition diagram. We generalize his observation:

\textbf{Lemma 4.7.1:}

Given a structured alphabet, the category \texttt{SLgraph} is \( \mathcal{E} - \mathcal{M} \)-factorizable with

(a) \( \mathcal{E} \) being the set of all epimorphisms of \texttt{SLgraph} and

(b) \( \mathcal{M} \) being the set of all label preserving graph monomorphisms.

\textbf{Proof:}

Let \( f : G \to H \) be a morphism in \texttt{SLgraph}. More precisely, we have \( f_E : E_G \to E_H \), \( f_V : V_G \to V_H \) and the labeling conditions

\[
(\forall v \in V_G)(l_{V_G}(v) \sqsubseteq l_{V_H}(f_V(v)))
\]
\[
(\forall e \in E_G)(l_{E_G}(e) \sqsubseteq l_{E_H}(f_E(e)))
\]

We can uniquely decompose the underlying graph morphism into an epimorphism and a monomorphism, and we label the intermediate graph with the labels of \( H \):

\[
G' = (f_E[E_G], f_V[V_G], s', t', l'_E, l'_V), \text{ where } s', t', l'_E, \text{ and } l'_V \text{ are the restrictions of } s_H, t_H, l_{E_H}, \text{ and } l_{V_H}, \text{ respectively.}
\]

Now, we decompose \( f \) into \( f = f_m \cdot f_e \):

\[
f_e = (f_E : E_G \to f_E[E_G], f_V : V_G \to f_V[V_G])
\]
\[
f_m = (\text{in}_{f_E[E_G]} : f_E[E_G] \to E_H, \text{in}_{f_V[V_G]} : f_V[V_G] \to V_H)
\]

where the morphism \( \text{in}_{f_E[E_G]} \) is the natural injection of \( f_E[E_G] \) into \( E_H \), etc. Trivially, \( f_e \) is an epimorphism satisfying the labeling conditions of an \texttt{SL}-graph morphism and \( f_m \) is a label preserving monomorphism. \( \square \)

This factorization satisfies the assumptions of Theorem 4.1.8, i.e., the property of being in \( \mathcal{E} \) or in \( \mathcal{M} \) is preserved by pushouts. The proof is left to the reader as an exercise (Exercise 4.8.13). Of course, you can define other decompositions also satisfying these assumptions. But, the advantage of our factorization is that it again restricts ambiguity to only one subdiagram, although the coproduct complement in \texttt{SLgraph} is ambiguous, but the factorization defined in Lemma 4.7.1 restricts coproduct constructions to \texttt{Lgraph}:

\textsuperscript{6}In their paper, Parisi-Presicce \textit{et al.} define the solution to be the maximal complement, since he uses the inverse relation.
**Lemma 4.7.2** (Decomposing pushouts in $\mathcal{SL}_{\text{graph}}$ [98]):
Decomposing the construction of a pushout complement in $\mathcal{SL}_{\text{graph}}$ according to Corollary 4.1.9 again restricts ambiguity to the subdiagram consisting only of $E$-morphisms.

Proof:

The assertion holds true for the underlying graphs. Consider the simplified diagram on top of the next page. More precisely, we had to consider $p_{Ee}, p_{Ve}, p_{Em}, p_{Vm}$, etc., i.e., all formulae hold for edges as well as for nodes. For reasons of simplicity, we omit this distinction. In subdiagrams (2) and (4), the construction of coproduct complements is applied to label preserving morphisms, and $\bar{p}_e$ and $\bar{p}_m$ are unambiguous because of the universal property of coproducts. In subdiagram (1), we first construct the coproduct complement $\bar{m}$ of a label preserving morphism. But then, we consider $\bar{m}' := \bar{g}_e \cdot \bar{m}$. Although $\bar{g}_e$ need not preserve labels, the composition does. Consider, e.g., a node $v$ in $B$ such that the label of $\bar{g}_e(v)$ is different from the label of $v$. Since the resulting diagram must be a pushout, this means that $v$ is an interface node and changing its label is caused by the label of the corresponding node in the graph $G'$. Therefore, $v$ cannot be in $\bar{B}'$, i.e., $\bar{m}'$ is label preserving, and there exists a unique $p_m'$.

This means that the pushout complements in subdiagrams (1), (2), and (4) are unambiguous if they exist. \(\Box\)

What about subdiagram (3)? Lemma 4.3.4 ensures existence of the maximal pushout complement. With respect to the underlying graphs, we have already discussed how to find the other solutions. How to find the labels?

**Lemma 4.7.3:**

If the diagram

\[
\begin{array}{c}
I \xrightarrow{p_e} B' \\
\downarrow{g_e} \\
C' \xrightarrow{p'_e} C''
\end{array}
\]

is a pushout diagram in $\mathcal{SL}_{\text{graph}}$ with $p_e$ and $g'_e$ being epimorphisms, the label $l_{C'}(y)$ of a node or an edge $y$ of $C'$ must satisfy:
(a) \( l_I(v) \subseteq l_{C'}(y) \subseteq l_{G'}(g_e'(p_e(v))) = l_{G'}(p_e'(y)) \) for all \( v \) with \( g_e(v) = y \) and
(b) \( l_{G'}(p_e'(y)) = \text{lub}(\{l_{C'}(y') \mid p_e'(y') = p_e'(y)\}) \cup \{l_B(y'') \mid g_e(y'') = p_e'(y)\} \).

An equivalent formulation of condition (a) is

\((a') \) \( \text{lub}\{l_I(v) \mid g_e(v) = y\} \subseteq l_{C'}(y) \subseteq l_{G'}(p_e'(y)) \)

Proof:

Since \( g_e \) is an epimorphism, \( C' \) does not contain any elements that are not images of elements of \( I \). We get condition (a) by the definition of morphisms in \( \mathcal{SL}_{\text{graph}} \) and the commutativity of the diagram. Condition (b) is a consequence of the pushout construction in \( \mathcal{SL}_{\text{graph}} \). \( \square \)

For the nodes and the edges of \( C' \), we can choose any labels that satisfy these conditions. Trivially, the maximal solution \( l_{C'}(y) := l_{G'}(p_e'(y)) \) does.

In the case of a production with an injective left-hand side, the underlying graph morphisms of \( p_e \) and \( p_e' \) are bijective, and the underlying graph of \( C' \) is identical to that of \( G' \) and therefore, it is unambiguous if the identification condition and the dangling condition are satisfied. In this case, ambiguity can arise only from the labeling. But, the condition 4.7.3(b) becomes simpler:

**Corollary 4.7.4:**

If in Lemma 4.7.3, \( p_e \) and \( p_e' \) are bijective graph morphisms, the label \( l_{C'}(y) \) of a node or an edge \( y \) of \( C' \) must satisfy:

(a) \( \text{lub}\{l_I(v) \mid g_e(v) = y\} \subseteq l_{C'}(y) \subseteq l_{G'}(p_e'(y)) \) and
(b) \( l_{G'}(p_e'(y)) = \text{lub}(\{l_{C'}(y)\} \cup \{l_B(y'') \mid g_e(y'') = p_e'(y)\}) \).

This corollary considers only subdiagram (3). But, we can rewrite it with respect to the whole production, since the morphisms of \( \mathcal{M} \) are label preserving, and the underlying graph morphisms of \( p_e, p_e', \) and \( \bar{p}_e \) are bijections:

**Corollary 4.7.5 (Pushout complements of injective productions):**

If the left-hand side of an \( \mathcal{SL}_{\text{graph}} \)-production is injective, the pushout complement \( C \) exists if the identification condition and the dangling condition are satisfied. The label \( l_{C'}(y) \) of a node or an edge \( y \) of \( C \) must satisfy:

(a) \( \text{lub}\{l_I(v) \mid g(v) = y\} \subseteq l_{C}(y) \subseteq l_{G}(\bar{p}(y)) \) and
(b) \( l_{G}(\bar{p}(y)) = \text{lub}(\{l_{C}(y)\} \cup \{l_B(y'') \mid \bar{g}(y'') = \bar{p}(y)\}) \).

Parisi-Presicce et al. have called the solutions \( l_{C'}(y) \) to this equation \( g_e' \)-complements. In their definition of derivability, they impose an additional constraint on the labeling. This condition is based on studying the set-theoretical details of the \( g_e' \)-complements. In our terminology, this additional property ensures existence of a unique minimal pushout complement which is used to complete the right-hand side of the derivation step. If we know this minimal pushout complement, Lemma 4.3.6 tells us how to find all the other solutions.

If we additionally assume the handle to be injective, the conditions become still simpler, since in the subdiagram of interest, the underlying graph morphisms \( p_e \) and \( g'_e \)
as well \( g_e \) and \( p_e' \) are bijections:

\[
l_I(v) \sqsubseteq l_C(y) \sqsubseteq l_G'(g_e(p_e(v)))
\]

\[
l_G'(p_e'(y)) = \lub\{l_C(y), l_B'(p_e(v))\}
\]

We can put these conditions together saying that each solution \( x \) of

\[
l_I(v) \sqsubseteq x \sqsubseteq \lub\{x, l_B'(p_e(v))\} = l_G'(p_e'(y))
\]

is a possible label of \( y \). The following diagram makes clear what happens:

\[
\begin{array}{c}
l_I(v) \rightarrow \boxed{\subseteq} \rightarrow l_B'(p_e(v)) \\
| \quad | \\
x \quad \boxed{\subseteq} \rightarrow \boxed{\subseteq} \rightarrow l_G'(p_e'(y))
\end{array}
\]

Since the structured alphabet can be considered a category with \( a \sqsubseteq b \) to be the (unambiguous) morphism from \( a \) to \( b \) and with the least upper bound as the pushout construction (Example 2.5.13), we have to find the pushout complements in the category of the structured alphabet.

As before, we do not formulate this result with respect to subdiagram (3), but with respect to the whole diagram. This is no problem, since the monomorphic part of the decomposition is label preserving:

**Lemma 4.7.6** (Pushout complements in the case of injections):

We assume both a production with an injective left-hand side and a derivation step with an injective handle such that the pushout complement \( C \) in \( \mathbf{Graph} \) exists. Then, we can use any pushout complement of

\[
\begin{array}{c}
l_I(g^{-1}(y)) \rightarrow \boxed{\subseteq} \rightarrow l_B'(p'(g^{-1}(y))) \\
| \quad | \\
l_C(y) \quad \boxed{\subseteq} \rightarrow \boxed{\subseteq} \rightarrow l_G'(p'(y))
\end{array}
\]

PO

to label a node (or an edge) \( y \) of \( C \) that has a pre-image in the interface graph, and \( l_C(y) = l_G'(p'(y)) \) if it has not.\(^7\)

We return to the Petri net examples to illustrate the results we have found.

**Example 4.7.7**:

On top of the next page, you see the derivation step we have constructed intuitively in Example 3.6.2. Let us consider the input place. The minimal solution is to label it with \( \{p_5\} \). Similarly, you can treat the other places. The detailed construction is given on the page 144.

But the input place can also be labeled with \( \{p_2, p_5\} \). This is a pushout complement, too. The reader can easily verify this by considering subdiagram (3) and adjusting subdiagram (4). In this case, however, \( p_2 \) also occurs on the right-hand side. This does not agree with the semantics of Petri nets. Although our

\(^7\)Please note that this is the inverse of a more general theorem we have proved some years ago [91, Theorem 2.11].
Example 4.7.7: Derivation step

definition of a derivation step allows arbitrary pushout complements as long as the right-hand side can be constructed with valid labels, it makes sense in many application areas to restrict the definition to the minimal pushout complement.

In Example 3.6.1, we have also constructed a pushout complement intuitively. It is left to the reader to show that this intuitive solution is the minimal one and that the other solutions lead to invalid labelings on the right-hand side (Exercise 4.8.18).

In the case of Petri nets, both sides of the productions describing firing a transition are injective graph morphisms. Furthermore, we have an application condition: The handle must be injective, too. This ensures an unambiguous pushout complement of the underlying graph morphisms. Furthermore, the structures of the alphabets used in Example 3.6.1 (Producer-Consumer Net) as well as in Example 3.6.2 (Dining Philosophers) allow to define a minimal solution. In both cases, this minimal solution coincides with the usual semantics of Petri nets.

In the case of non-injective handles, the situation is more complicated, but it can be effectively treated if we restrict the structure of the alphabet suitably [98].
Example 4.7.7: Detailed construction of the pushout complement
4.8 Exercises

Exercise 4.8.1 (Diagonalization property):
Let $\mathcal{K}$ be an $\mathcal{E}-\mathcal{M}$-factorizable category. If we have a commutative diagram
\[ \begin{array}{ccc}
g \cdot e = m \cdot f \\
\downarrow \quad \downarrow \\
e & d & g
\end{array} \]
with $e \in \text{Mor}_E$, $m \in \text{Mor}_M$, and arbitrary morphisms $f, g \in \text{Mor}_K$, then there exists a unique $d \in \text{Mor}_K$ with $f = d \cdot e$ and $g = m \cdot d$.

Exercise 4.8.2:
A graph morphism $f = (f_E, f_V) : G_1 \to G_2$ can be factorized component by component in $\text{Set}$: $f_E = m_E \cdot e_E$ and $f_V = m_V \cdot e_V$. Consider the diagram
\[ \begin{array}{ccc}
G_1E & \overset{e_E}{\longrightarrow} & H_E & \overset{m_E}{\longrightarrow} & G_2E \\
\downarrow s_1 & & \downarrow s_{\mathcal{H}} & & \downarrow s_2 \\
G_1V & \overset{e_V}{\longrightarrow} & H_V & \overset{m_V}{\longrightarrow} & G_2V
\end{array} \]
Prove that $\text{Graph}$ is $\mathcal{E}-\mathcal{M}$-factorizable by showing that morphisms $s_{\mathcal{H}}$ and $t_{\mathcal{H}}$ can be defined such that $(e_E, e_V)$ and $(m_E, m_V)$ are graph morphisms.

Exercise 4.8.3:
Theorem 4.1.8 assumes that the $\mathcal{E}-\mathcal{M}$-factorizable category $\mathcal{K}$ has pushouts, i.e., that we can construct a pushout diagram over each pair of morphisms. We can weaken this assumption. Prove: If $\mathcal{K}$ is $\mathcal{E}-\mathcal{M}$-factorizable with $\mathcal{M}$ being a set of coretractions, then we can decompose a given pushout diagram as depicted in the theorem.

Exercise 4.8.4:
Assume that $\mathcal{K}$ has coproduct and initial object and consider the following subset of morphisms:
\[ \text{Mor}_{C(\mathcal{K})}(A, B) := \{ f \in \text{Mor}_K(A, B) \mid f \text{ has coproduct complement} \} \]
Prove that $C(\mathcal{K}) := (\text{Obj}_K, \langle \text{Mor}_{C(\mathcal{K})}(A, B) \rangle, \cdot_K)$ is a subcategory of $\mathcal{K}$.

Exercise 4.8.5:
Prove: If in the pushout diagram
\[ \begin{array}{ccc}
I & \overset{p}{\longrightarrow} & B \\
\downarrow \quad \downarrow d & & \downarrow \quad \downarrow \bar{d} \\
D & \overset{d \cdot p}{\longrightarrow} & G
\end{array} \]
$\bar{d}$ is an epimorphism, then it is an isomorphism.

Exercise 4.8.6:
We modify Example 4.3.3 by removing element 4 from $I$:
As in Example 4.3.3, we can construct the coproduct complements \( \bar{m} \) and \( \bar{p}_m \):

\[
\begin{align*}
I : \{1, 3\} & \xrightarrow{p_m} B : \{1, 3, 4, 5\} \\
\bar{g}_e & \downarrow \\
G : \{1, [3, 4], 5\} & \leftarrow \bar{m}
\end{align*}
\]

Prove that it is impossible to find a morphism \( g_e \) such that \( \bar{p}_m \cdot g_e = \bar{g}_e \cdot p_m \) commutes!

**Exercise 4.8.7:**

Construct the pushout complement of the following graph morphisms:

\[
\begin{array}{c}
I \xrightarrow{p} B \\
\end{array}
\]

**Exercise 4.8.8:**

Consider the following morphisms \( p \) and \( \bar{g} \). Both \( p_E \) and \( p_V \) are not injective.

**Exercise 4.8.9:**

Construct the pushout complements by applying Lemma 4.5.6. Can you find another pushout complement?
Reconsider Exercise 3.8.6. Now, you should be able to construct the solution systematically.

**Exercise 4.8.10:**
Consider the following situation:

Construct the pushout complements following Theorem 4.5.8. Describe the various ways to define $r$.

**Exercise 4.8.11:**
Consider the following morphisms in $\mathcal{L}_{\text{graph}}$:

Construct all the pushout complements for nodes and edges, separately. Which combinations of these solutions lead to pushout complements in $\mathcal{L}_{\text{graph}}$ and which do not?

**Exercise 4.8.12:**
Reconsider Example 3.3.5. We have applied the productions \( p_6, p_5, \) and \( p_3 \) to construct a derivation, but we have done this intuitively. Verify the derivation steps by explicitly constructing the context hypergraphs and by checking the conditions you need.

**Exercise 4.8.13:**
Prove that the decomposition defined in Lemma 4.7.1 satisfies the assumptions of Theorem 4.1.8.

**Exercise 4.8.14:**
Consider the term rewriting rule \( m(i(x), x) \rightarrow e \), i.e., multiplying an element by its inverse yields the unit element. Design

(a) a hypergraph production analogous to Example 3.3.5,
(b) an SL-graph production analogous to Example 3.7.4

such that derivations steps generated by these productions correspond to the simplification defined by the rewriting rule.

**Exercise 4.8.15:**
The following exercises return to Example 3.7.4:

\[
\begin{array}{c}
B^l \\
\begin{array}{c}
\text{op}^1 \\
\text{op}^2 \\
1 \\
x^3 \\
2 \\
y^4 \\
\end{array}
\end{array}
\begin{array}{c}
I \\
\begin{array}{c}
\text{op}^1 \\
\text{op}^2 \\
x^3 \\
y^4 \\
\end{array}
\end{array}
\begin{array}{c}
B^r \\
\begin{array}{c}
\text{op}[1, 2] \\
1 \\
x^3 \\
2 \\
y^4 \\
\end{array}
\end{array}
\end{array}
\]

The left-hand side of this production is a monomorphism in \( SL\text{graph} \). First, we have intuitively applied it to the following graph using an injective handle:

\[
\begin{array}{c}
G^l_1 \\
\begin{array}{c}
\ldots \\
\text{op}^1 \\
\text{op}^2 \\
1 \\
2 \\
x^3 \\
\text{op}[1, 2] \\
\end{array}
\end{array}
\]

Construct the pushout complement systematically according to the decomposition theorem and define its labeling according to Lemma 4.7.6.

**Exercise 4.8.16:**
Now, consider the noninjective handle necessary to derive the following \( G^l_2 \):
Construct the pushout complement systematically according to the decomposition theorem and examine its labeling.

Exercise 4.8.17:
What about the following $G_3^l$:

Is it possible to construct a pushout complement? (Please note that there is no common subexpression!) If so, what happens on the right-hand side?

Exercise 4.8.18:
In Example 3.6.1, we have discussed a producer-consumer net. The derivation step describing firing transition $t_3$ is given on top of the next page. Examine this derivation step systematically:

(a) Construct the pushout complement $C$ in a detailed way according to the decomposition.

(b) Show that the solution $C$ we have intuitively given is the minimal solution.

(c) Construct some other pushout complements $C_i$ and show that these complements do not result in valid derived graphs $G_i^r$. 

## Contents

4 Effectively Constructing Derivation Steps 107

4.1 Pushout Complements and Their Decomposition . . . . . . . . . . . . . 108
4.2 Pushout Complements Using Coproduct Complements . . . . . . . . . 114
4.3 The General Case . . . . . . . . . . . . . . . . . . . . . . 118
4.4 Constructing Pushout Complements in Set . . . . . . . . . . . . . . . . 123
4.5 The Situation in Graph . . . . . . . . . . . . . . . . . . . . . 126
4.6 Labeled Graphs and Hypergraphs . . . . . . . . . . . . . . . . . . . . 136
4.7 Structurally Labeled Graphs . . . . . . . . . . . . . . . . . . . . . . . 138
4.8 Exercises . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145