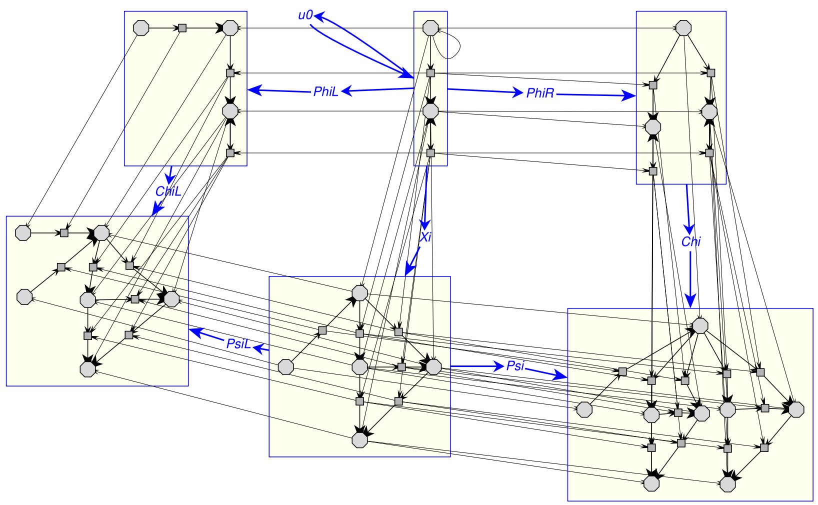
# A Relation-Algebraic Approach to Graph Structure Transformation

WOLFRAM KAHL




Bericht Nr. 2002-03
June 2002

Universität der Bundeswehr München
Fakultät für
INFORMATIK
Werner-Heisenberg-Weg 39 • 85577 Neubiberg • Germany

# A Relation-Algebraic Approach to Graph Structure Transformation

WOLFRAM KAHL[1]

**Abstract**

Graph transformation is a rapidly expanding field of research, motivated by a wide range of applications.

Such transformations can be specified at different levels of abstraction. On one end, there are "programmed" graph transformation systems with very fine control over manipulation of individual graph items. Different flavours of rule-based graph transformation systems match their patterns in more generic ways and therefore interact with the graph structure at slightly higher levels.

A rather dominant approach to graph transformation uses the abstractions of category theory to define matching and replacement almost on a black-box level, using easily understandable basic category-theoretic concepts, like pushouts and pullbacks. However, some details cannot be covered on this level, and most authors refrain from resorting to the much more advanced category-theoretic tools of topos theory that are available for graphs, too — topos theory is not felt to be an appropriate language for specifying graph transformation.

In this thesis we show that the language of relations can be used very naturally to cover all the problems of the categoric approach to graph transformation. Although much of this follows from the well-known fact that every graph-structure category is a topos, very little of this power has been exploited before, and even surveys of the categoric approach to graph transformation state essential conditions outside the category-theoretic framework.

One achievement is therefore the capability to provide descriptions of all graph transformation effects on a suitable level of abstraction from the concrete choice of graph structures.

Another important result is the definition of a graph rewriting approach where relational matchings can match rule parameters to arbitrary subgraphs, which then can be copied or deleted by rewriting. At the same time, the rules are still as intuitive as in the double-pushout approach, and there is no need to use complicated encodings as in the pullback approaches.

In short: A natural way to achieve a double-pushout-like rewriting concept that incorporates some kind of "graph variable" matching and replication is to amalgamate pushouts and pullbacks, and the relation-algebraic approach offers appropriate abstractions that allow to formalise this in a fully component-free yet intuitively accessible manner.

[1]Current e-mail address: kahl@mcmaster.ca. This habilitation thesis was completed during the author's appointment at the Institute for Software Technology of Universität der Bundeswehr München.

## Acknowledgements

Although my interest in graphs and structures may have been "hard-wired" in my brain, it was Gunther Schmidt who led me on the one hand to applications of graph transformation in software development, and on the other hand to relation-algebraic formalisations in general, and to their applications to graphs in particular. Besides guiding me with his deep understanding of the calculus of relations as a valuable tool for formalisation and reasoning, he also alerted me to the importance of being aware of the axiomatic basis underlying one's work, and refraining from assuming more than necessary for the task at hand.

Another important influence has been Yasuo Kawahara, who introduced me to Dedekind categories and to working in weaker axiomatisations of relation categories, and whose pioneering work on "pushout-complements in toposes" provided an important stepping stone for the current thesis.

I wish to thank many friends and colleagues for discussions on more or less directly related topics that considerably helped to shape my ideas, let me only mention Andrea Corradini, Jules Desharnais, Hitoshi Furusawa, Fabio Gadducci, Hélène Jacquet, Bernhard Möller, John Pfaltz, and Michael Winter.

Finally, I am infinitely grateful to my wife Liping and daughter Cynthia for their patience, understanding and support during the genesis of this thesis.

# Contents

**1 Introduction** **1**
1.1 Graph Grammars and Graph Transformation . . . 1
1.2 The Categoric Approaches to Graph Transformation . . . 3
1.2.1 Pushouts . . . 4
1.2.2 More About Pushouts in Graphs . . . 7
1.2.3 The Double-Pushout Approach . . . 7
1.2.4 The Gluing Condition . . . 9
1.2.5 Restricting Derivations . . . 12
1.2.6 The Single-Pushout Approach . . . 13
1.2.7 Shortcomings of the Pushout Approaches . . . 15
1.2.8 The Pullback Approach . . . 15
1.3 Graph Transformation with Relational Matching . . . 21
1.4 Relation Algebras and Generalisations . . . 24
1.5 Contrasting the Relational and the Categoric Approaches . . . 26
1.6 Overview . . . 27
1.7 Eddi . . . 29

**2 Graph Structures and Their Parts** **30**
2.1 Preliminaries: Sets, Lattices . . . 31
2.2 Signatures . . . 34
2.3 Algebras and Subalgebras . . . 36
2.4 Subalgebras in Graph Structures . . . 41
2.5 Partial Algebras and Weak Partial Subalgebras . . . 44
2.6 Pseudo-Complements . . . 46
2.7 Semi-Complements . . . 50
2.8 Naïve Graph Rewriting . . . 56
2.9 Discreteness in Graph Structures . . . 57
2.10 Coregular Parts and Base Elements . . . 63

**3 Allegories of $\Sigma$-Algebras** **66**
3.1 Preliminaries: Categories and Allegories . . . 66
3.2 Abstract $\Sigma$-Algebras and Relational Homomorphisms . . . 73
3.3 Constructions in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ . . . 77

**4 Dedekind Categories of Graph Structures** **84**
4.1 Preliminaries: Distributive Allegories and Dedekind Categories . . . 84
4.2 Joins in General $\Sigma$-Algebra Allegories . . . 88
4.3 Relational Homomorphisms Between Graph Structures . . . 90
4.4 Pseudo- and Semi-Complements in $\Sigma\text{-}\mathsf{GS}_{\mathbf{D}}$ . . . 93
4.5 Constructions in $\Sigma\text{-}\mathsf{GS}_{\mathbf{D}}$ . . . 94
4.6 Discrete Relations . . . 95

**5 Categoric Rewriting in a Relational Setting** **98**
5.1 Pullbacks . . . 99
5.2 Transitive and Difunctional Closures . . . 102
5.3 Pushouts . . . 103
5.4 Pushout Complements . . . 107
5.5 Pullback Complements . . . 116
5.6 Pushouts of Partial Functions . . . 120
5.7 Summary . . . 129

**6 Relational Rewriting in Dedekind Categories** **131**
6.1 Gluing Setup . . . 131
6.2 Amalgamating Pushouts and Pullbacks to Pullouts . . . 134
6.3 Pullout Complements . . . 138
6.4 Pullout Rewriting . . . 140
6.5 The Weak Pullout Construction . . . 143

**7 Conclusion and Outlook** **151**

**Bibliography** **153**

**A Proofs of Auxiliary Properties** **165**
A.1 Allegory Properties . . . 165
A.2 Partial Identities . . . 167
A.3 Dedekind Category Properties . . . 169
A.4 Semi-Complements and Partial Identities in Dedekind Categories . . . 170
A.5 Symmetric Quotients . . . 172

**B Proofs for Relational Rewriting, Chapter 6** **173**
B.1 Correctness of the Glued Tabulation Construction . . . 173
B.2 Correctness of the Pullout Complement Construction . . . 179
B.3 Monomorphy of Weak Pullouts . . . 184
B.4 Correctness of the Direct Result Construction . . . 190
B.5 The Straight Host Construction Yields Weak Pullout Complements . . . 199

**Index** **201**

# Chapter 1

# Introduction

We start this introductory chapter with an overview over graph transformation, followed by a more detailed presentation of the categoric approaches to graph transformation. In Sect. 1.3 we sketch the problem of relational matching and shortly sketch our solution. All this serves as introduction into the problem area of the present thesis.

The solutions rely on axiomatisations of categories of relations, and we present some historical background for this in Sect. 1.4. Although categories of relations are still categories, the style of relational formalisations is frequently very different from the style of category-theoretic formalisations; we discuss this effect in Sect. 1.5.

Finally, we present an overview over the remainder of this thesis in Sect. 1.6.

## 1.1 Graph Grammars and Graph Transformation

Research on graph grammars started in the late 1960s [PR69, Sch70, EPS73] and has since evolved into a very active area, as documented by a gradually increasing number of conference series devoted to the topic [CER78, ENR82, ENRR87, FK87, EKR91, CR93b, SE93, C$^{+}$96, EEKR00, ET00].

Given the fact that graphs are a useful means for modelling many different kinds of complex systems, graph grammars and graph replacement was, from the beginning, strongly motivated by applications in computer science, such as networks, and also in biology, such as modelling the growth of plants [LC79, PL90].

Important areas of applications currently include applied software systems and systems modelling [Jon90, Jon91, Jon92, EB94], many aspects of parallelism, distributed systems and synchronisation mechanisms [BFH85, Kre87, KW87, Sch90], implementation of term rewriting and operational semantics of functional (and logic) programming [Pad82, HP91, CR93a, PvE93, SPvE93, Kah96, Plu99, BS99], operational semantics of object oriented languages [WG96, WG99], and many aspects of visual programming languages [BMST99].

A survey of the foundations of graph transformation is contained in the first volume [Roz97] of a three-volume handbook. The second volume [EEKR99] concentrates on applications, languages, and tools, and there already has been a first international workshop on "Applications of Graph Transformations with Industrial Relevance" [NSM00].

In recent years, there has been a shift of attention, accompanied by a shift of terminology, from "graph grammars" to "graph transformation", from "productions" to "(transformation) rules", as problems of describing, generating, and parsing classes of graphs (as "graph languages") are nowadays overshadowed by problems of using graph transformation to specify or describe complex system behaviour and step-wise adaptation or refinement of systems or abstract entities represented by graphs.

Accordingly, we shall use the following nomenclature for the essential items involved in an individual graph transformation step:

- A *rule* usually contains at least a *left-hand side* and a *right-hand side*, which usually both are graphs, and some indication of how instances of the right-hand side are going to replace matched instances of the left-hand side.
- Rule application to some *application graph* requires identification of a *redex* in the application graph, usually via some kind of *matching* from the rule's left-hand side to the application graph.
- Performing the *transformation* then produces a *result graph*, and usually it is possible to identify, via a *result embedding*, some instance of the rule's right-hand side in the result graph.

$$\begin{tikzcd}
\text{left-hand side} \arrow[r, "\textit{rule}"] \arrow[d, "\textit{redex / matching}"'] & \text{right-hand side} \arrow[d, "\textit{result embedding}"] \\
\text{application graph} \arrow[r, "\textit{transformation}"] & \text{result graph}
\end{tikzcd}$$

Traditionally, there are two main groups of approaches to graph transformation:

- The "set-theoretic" approaches consider graphs concretely as made up of a set of vertices and a set of edges between vertices, and allow rules to describe more or less arbitrary operations on graphs. Two important classes of graph grammars belong here:
  - *Node replacement graph grammars* have rules that specify replacement graphs for single nodes, and how these replacement graphs are connected into the application graph depending on how the replaced node was connected. Through such a replacement step, edges may be deleted or created in numbers that cannot (and need not) be controlled by the rules. For an introduction, see [ER97].
  - *Edge replacement graph grammars*, or, more usually, *hyperedge replacement graph grammars* have rules that specify a replacement graph for a single (hyper-)edge, and which nodes of the replacement graph should be identified with the nodes the replaced edge was incident with. Such a replacement step cannot delete nodes; it deletes a single edge and creates new nodes and edges in numbers completely specified by the applied rule. For an introduction, see [DKH97].
- In the "algebraic" approaches, graph homomorphisms make up the rules and serve as matchings, and category-theoretic concepts of pushout and pullback are used to specify rewriting.

  As a result, larger patterns (not single items as in the set-theoretic approaches) can be rewritten, and reasoning about graph transformation is facilitated via the category-theoretic abstraction underlying these approaches. If this was followed through consequently, then nodes and edges should never occur in arguments about categoric graph transformation — proofs should be completely *component-free*.



Programmed graph transformation [Sch97, SWZ99] has historically been closer to the set-theoretic approaches.

We are interested mainly in the categoric approaches, which we therefore present in more detail in the next section.

## 1.2 The Categoric Approaches to Graph Transformation

The so-called "algebraic approach to graph transformation" is really a collection of approaches that essentially rely on category-theoretic abstractions.

Historically, Ehrig, Pfender and Schneider developed the double-pushout approach as a way to generalise Chomsky grammars from strings to graphs, using pushouts as "gluing construction" to play the rôle of concatenation on strings [EPS73]. The name "algebraic approach" derives from the fact that graphs can be considered as a special kind of algebras, and that the pushout in the appropriate category of algebras was perceived more as a concept from universal algebra than from category theory. The first three graph grammar conference proceedings contain early tutorial introductions [Ehr78, Ehr87, EKL91].

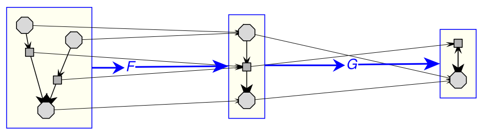
Motivated by the shortcomings of the double-pushout approach that was based on *total* graph homomorphisms, the end of the 1980s saw the emergence of the *single-pushout approach* based on categories of *partial* graph homomorphisms. The work of Raoult [Rao84] and Kennaway [Ken87, Ken91] was mainly motivated by applications to term graph rewriting, while Löwe [Löw90, LE91, Löw93] took a more general approach.

A recent exposition of the foundations of the pushout approaches can be found as chapters 3 and 4 [CMR+97, EHK+97] in the foundations volume of the "Handbook of Graph Grammars and Computing by Graph Transformation" [Roz97].

A less prominent categoric approach "turns all arrows around" — Bauderon proposed algebraic graph rewriting based on pullbacks [Bau97, BJ01]. In contrast to the pushout approaches, the pullback approach can handle deletion and duplication with ease. So far, only pullbacks of total graph homomorphisms have been considered, but nevertheless, single- and double-pullback rewriting can be used for rewriting concepts of different complexity.

The remainder of this section is a "guided tour" through the concepts and variants of the categoric approach to graph rewriting, starting with pushouts and the double-pushout approach, including the "restricting derivation" variant of the latter. We then continue with the single-pushout approach and finish with a short presentation of the pullback approach.

From now on, we will illustrate our arguments with diagrams involving graphs and graph morphisms. Here is an example of such a diagram:




It shows three graphs, each in a rectangular box, with two morphisms $F$ and $G$ in-between.

Each graph consists of octagonal *nodes*, small square *edges*, and for every edge a *source tentacle* from the edge to the source node of that edge — this is drawn as a small-tipped arrow *from the source node towards the edge* — and a *target tentacle* from the edge to the target node of that edge — this is drawn as a large-tipped arrow *from the edge towards the target node*.

This unconventional way to draw graphs makes it easier to make explicit the edge mappings of graph homomorphisms. Intuitively, the two tentacles together with the edge square may be understood as a single edge "from the source node to the target node"; this intuition motivates the directions of the arrows.

On the diagram level, graphs can be understood as nodes, and graph homomorphisms as edges, and they are presented in the same way: The homomorphism's name is connected via two thick tentacles with the homomorphism's source and target graphs, and both tentacles have arrows in the direction of the homomorphism.

The homomorphism itself is represented by thin arrows connecting graph *items* (nodes and edges) in the source graph with their images in the target graph.

A conventional *graph homomorphism* has to map every node to a single node, and every edge to a single edge, such that the source and target nodes of the image edge are the images of the source and target nodes of the original edge.

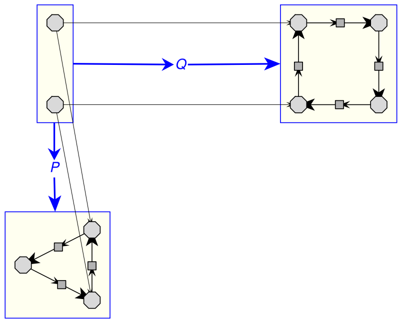
Therefore it is easy to check that $F$ and $G$ in the drawing above are indeed graph homomorphisms. The target of $G$ is a *unit graph*, that is, a graph consisting of a single edge and a single node which is both source and target of that edge. Such an edge where source and target coincide is usually called *loop*. In our drawings, the two tentacles of loop edges usually merge into a single two-tipped arrow, as above in the target of $G$.

### 1.2.1 Pushouts

Because of its importance in the context of categorical graph rewriting, and in view of its simplicity and intuitive accessibility, we start with a detailed introduction to the definition of the pushout construction, and provide step-by-step illustrations of an example in the category of unlabelled graphs (see Def. 5.3.1 for a compact definition of pushouts).

The setup for a pushout is a diagram consisting of two morphisms $P$ and $Q$ with common source $\mathcal{A}$.

$$\begin{tikzcd} \mathcal{A} \arrow[r, "Q"] \arrow[d, "P"'] & \mathcal{C} \\ \mathcal{B} & \end{tikzcd}$$

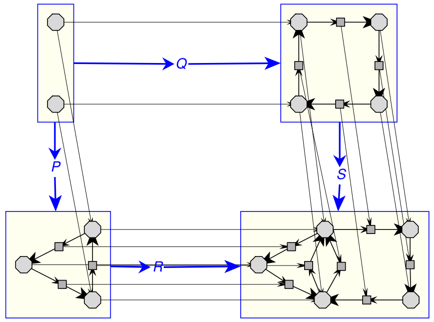



The *pushout for $P$ and $Q$* then consists of a *pushout object* $\mathcal{D}$ and two morphisms $R$ and $S$ with $\mathcal{D}$ as target, such that, first of all, the resulting square commutes, that is, all compositions of morphisms along different paths connecting the same objects are equal,



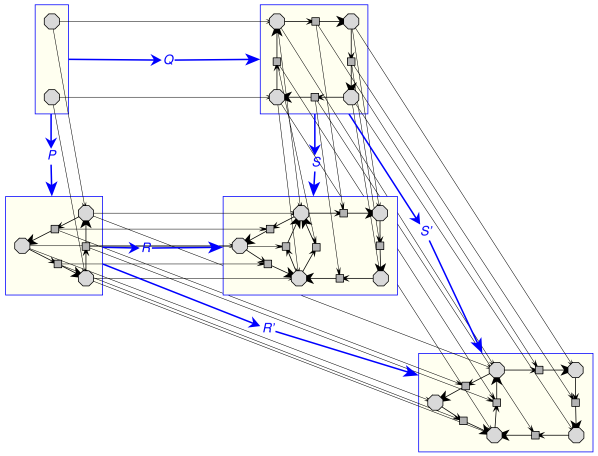
which in this case just means $P;R = Q;S$ (we use a small semicolon ";" for composition, and write composition in the diagrammatic order[1]).

$$\begin{tikzcd} \mathcal{A} \arrow[r, "Q"] \arrow[d, "P"'] & \mathcal{C} \arrow[d, "S"] \\ \mathcal{B} \arrow[r, "R"] & \mathcal{D} \end{tikzcd}$$




Now there may be more candidates for a commuting square completion, such as another object $\mathcal{D}'$ with morphisms $R'$ and $S'$ as in the following example:

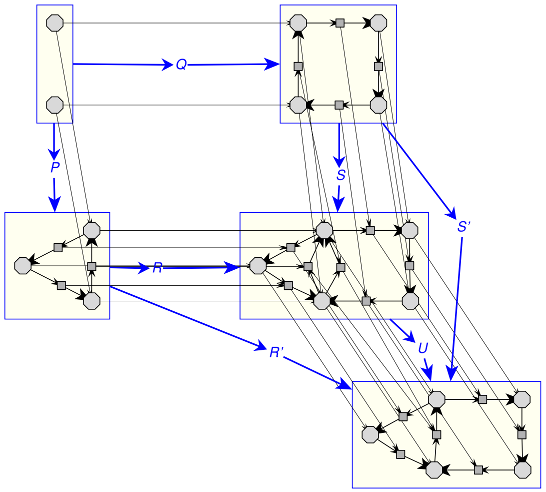
$$\begin{tikzcd} \mathcal{A} \arrow[r, "Q"] \arrow[d, "P"'] & \mathcal{C} \arrow[d, "S"] \arrow[ddr, "S'"] & \\ \mathcal{B} \arrow[r, "R"] \arrow[drr, "R'"'] & \mathcal{D} & \\ & & \mathcal{D}' \end{tikzcd}$$




The pushout proper is then defined by the following *universal characterisation*: For every such candidate for a commuting square completion, there has to exist a unique arrow $U$ from the pushout object $\mathcal{D}$ to the candidate object $\mathcal{D}'$ such that the candidate morphisms can be *factorised* via $U$, that is, they can be expressed as compositions of the pushout morphisms with this unique morphism $U$, which concretely means $R' = R;U$ and $S' = S;U$.

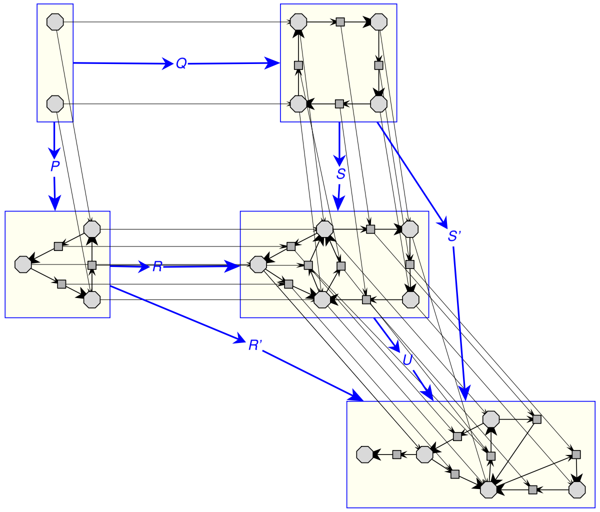
[1] that is, we compose $R : \mathcal{A} \to \mathcal{B}$ and $S : \mathcal{B}' \to \mathcal{C}$ to $R;S$ only if $\mathcal{B} = \mathcal{B}'$.

$$\begin{tikzcd} \mathcal{A} \arrow[r, "Q"] \arrow[d, "P"'] & \mathcal{C} \arrow[d, "S"'] \arrow[ddr, "S'"] & \\ \mathcal{B} \arrow[r, "R"] \arrow[drr, "R'"'] & \mathcal{D} \arrow[dr, dotted, "U"] & \\ & & \mathcal{D}' \end{tikzcd}$$




It is easy to see that in this example, there are morphisms $V_R$ and $V_S$ in the opposite direction, that is from $\mathcal{D}'$ to $\mathcal{D}$, such that $R = R' ; V_R$ and $S = S' ; V_S$, but there is no morphism $V$ from $\mathcal{D}'$ to $\mathcal{D}$ that would factorise *both* $R$ and $S$, since the edge that has two pre-images by $U$ can have only one image by $V$.

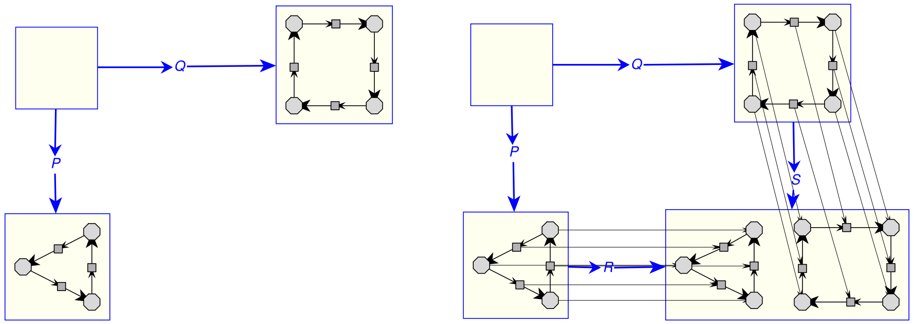
Other candidate commuting square completions are even "farther away" from being pushouts: In the following example, because of the additional identification of two vertices there is not even a morphism $V_S$ such that $S = S' ; V_S$. In addition, this example features graph items not covered by $R'$ or $S'$:



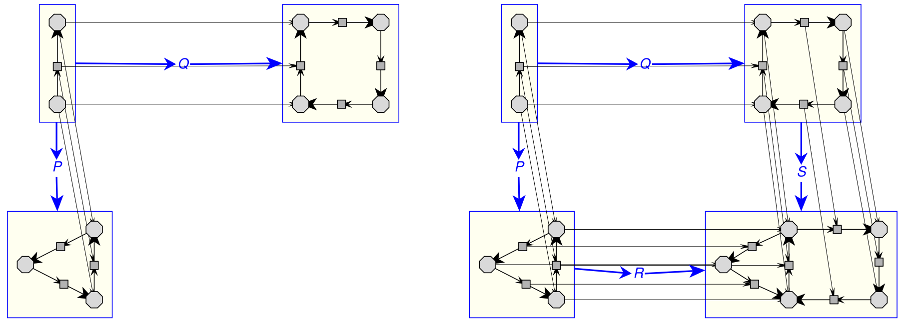



### 1.2.2 More About Pushouts in Graphs

The simplest case of pushouts arises if the graph $\mathcal{A}$ is empty, or, in general, if the object $\mathcal{A}$ is initial in the category under consideration. In this case, every choice of object $\mathcal{D}$ and arrows $R$ and $S$ to $\mathcal{D}$ makes the square commute, so the pushout condition degenerates to the condition for the disjoint sum (also called coproduct).




In general, pushouts may be understood as producing first a disjoint sum, and then *gluing* the two parts together along the *interface* indicated by $\mathcal{A}$ and its embeddings into the two constituent graphs.

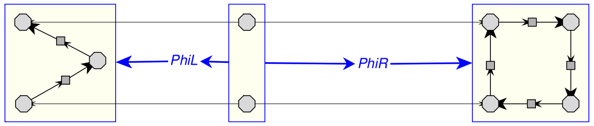



As shown in this example, $\mathcal{A}$ may contain edges. However, in the context of graph rewriting the interface usually just consists of the "real interface" between, for example, the redex and the context, and therefore is a discrete graph, that is, a graph consisting only of nodes. An example for that was given in the previous section — it differs from this example here only in that the edge has been taken away from the interface. Therefore, in the example of the last section, only the images of the two interface vertices were affected by the gluing, while the two edges connecting the respective images in $\mathcal{B}$ and $\mathcal{C}$ remained distinct.

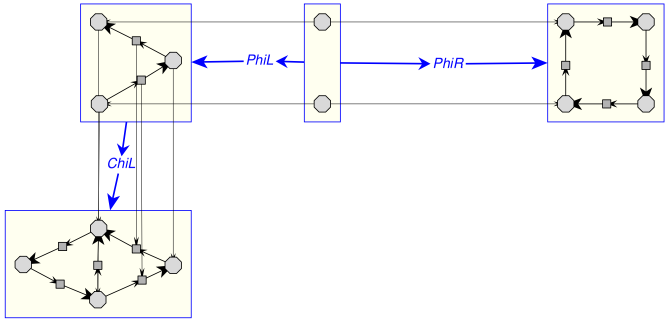
### 1.2.3 The Double-Pushout Approach

The *double-pushout approach* is the "classical" variant of the "algebraic approach" to graph rewriting, going back to [EPS73].

In this approach, a rewriting rule is a *span* $\mathcal{L} \xleftarrow{\Phi_L} \mathcal{G} \xrightarrow{\Phi_R} \mathcal{R}$ of morphisms, that is a pair of morphisms $\Phi_L$ and $\Phi_R$ starting from a common source $\mathcal{G}$, called the *gluing object*, and ending in the rule's *left-hand side* $\mathcal{L}$ and *right-hand side* $\mathcal{R}$, respectively. Usually, $\Phi_L$ must be injective (a detailed discussion of this issue is contained in [HMP01]). As an example consider the following graph rewriting rule, which deletes a path of length two between the two interface nodes, and inserts a four-node cycle with one edge connecting the interface nodes.




A *redex* for such a rule is a morphism $X_L : \mathcal{L} \to \mathcal{A}$ from the rule's left-hand side $\mathcal{L}$ into some *application graph* $\mathcal{A}$.




Application of the rule has to establish a *double-pushout diagram* of the following shape:

$$\begin{tikzcd}
\mathcal{L} \arrow[d, "X_L"'] & \mathcal{G} \arrow[l, "\Phi_L"'] \arrow[r, "\Phi_R"] \arrow[d, "\Xi"'] & \mathcal{R} \arrow[d, "X_R"'] \\
\mathcal{A} & \mathcal{H} \arrow[l, "\Psi_L"'] \arrow[r, "\Psi_R"] & \mathcal{B}
\end{tikzcd}$$

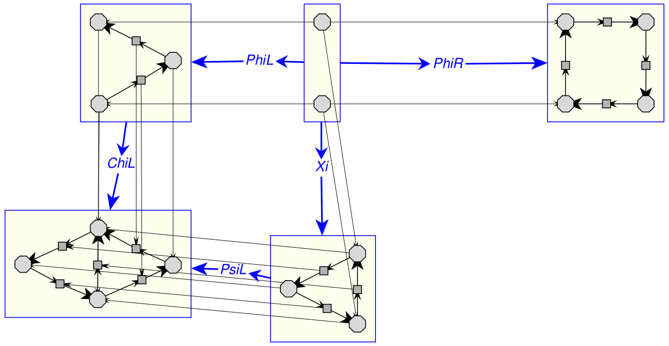
Here we encounter a first weak point of the categoric approach: While pushouts are a universal construction that may be described in very simple category-theoretic terms, the first step necessary for producing such a double-pushout diagram is *not* of this kind.

What is needed here is a *host object* $\mathcal{H}$ together with a *host morphism* $\Xi$ from the gluing object to the host object such that the matching can be reconstructed as part of the pushout for $\Phi_L$ and $\Xi$. This is called a *pushout complement* for $\Phi_L$ and $X_L$.

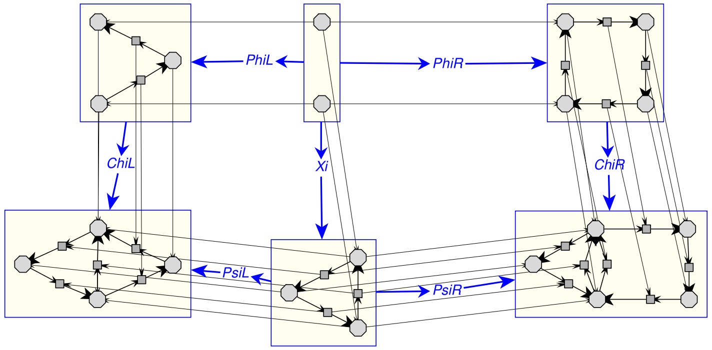
The fundamental result enabling the success of the double-pushout approach is that, for graphs, there is a simple condition on $\Phi_L$ and $X_L$, called the *gluing condition*, that is necessary and sufficient to ensure that the pushout complement in the category of graphs exists — more about the gluing condition below.

For the time being, it is sufficient to know that in our example, the gluing condition holds. A pushout complement is easily constructed as a subgraph of the application graph:





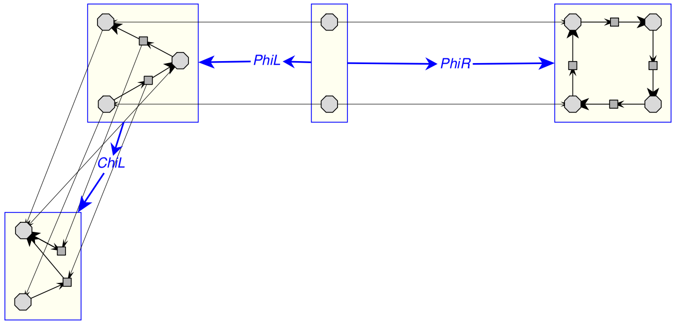

The pushout shown in 1.2.1 then completes the double-pushout diagram:



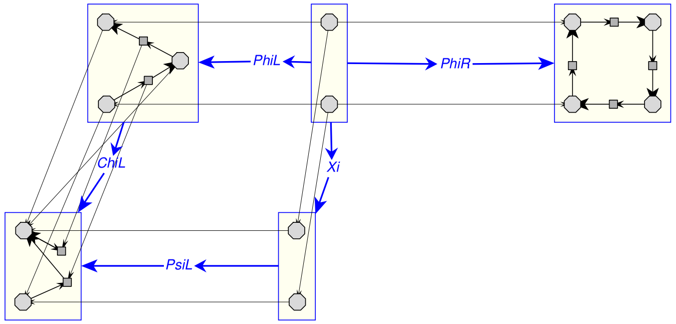

Such a rewriting step is usually called a *direct derivation* of $\mathcal{B}$ from $\mathcal{A}$. If $r$ is an identifier denoting the rule $\mathcal{L} \xleftarrow{\Phi_L} \mathcal{G} \xrightarrow{\Phi_R} \mathcal{R}$, then such a direct derivation via $r$ is frequently written $\mathcal{A} \overset{r}{\Rightarrow} \mathcal{B}$.

### 1.2.4 The Gluing Condition

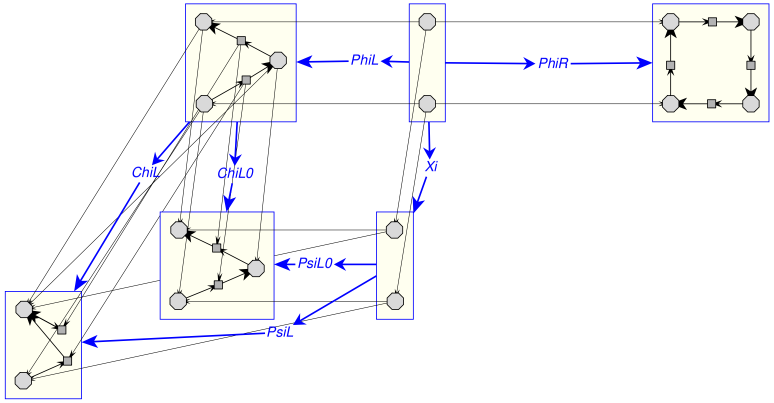
For our rule from above, consider the following matching:



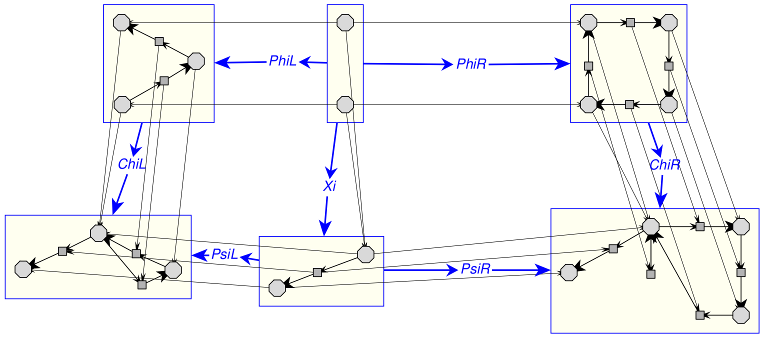
Since this matching morphism does not add any context, the host morphism can only be the identity on the gluing graph.




However, this host is not a pushout complement: The pushout of $\Phi_L$ and $\Xi$ is the identity on the left-hand side and therefore does not reconstruct the identification that is part of the original matching $X_L$ from the left-hand side to the application graph:




In contrast, if the identification in the matching affects only interface nodes, then there is no problem. In the following example, the host is a proper pushout complement:

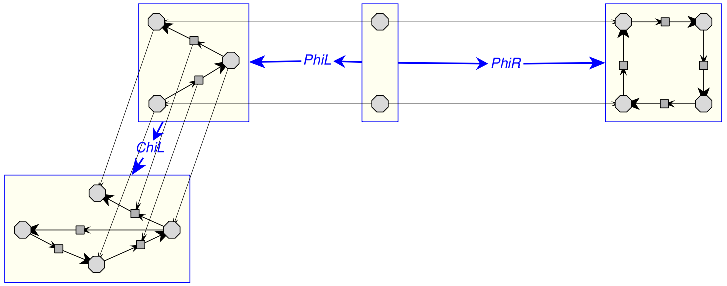




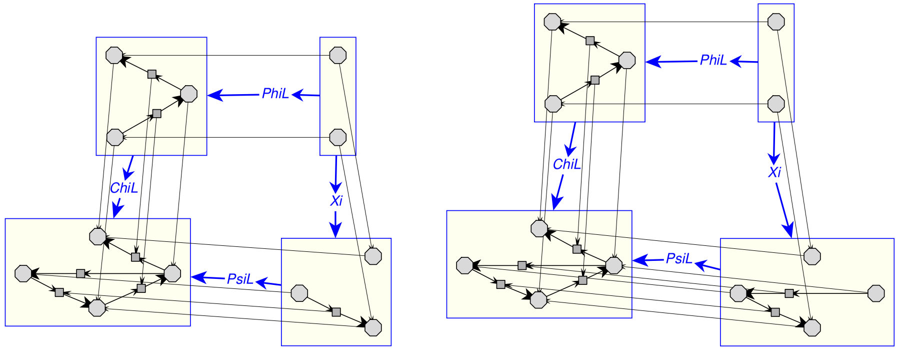

In general, for the existence of a pushout complement it is therefore necessary that the following is satisfied:

**Definition 1.2.1** For two graph morphisms $\Phi : \mathcal{G} \to \mathcal{L}$ and $X : \mathcal{L} \to \mathcal{A}$, the *identification condition* holds iff whenever two different nodes of $\mathcal{L}$ are identified by X, then they both lie inside the image of $\Phi$. □

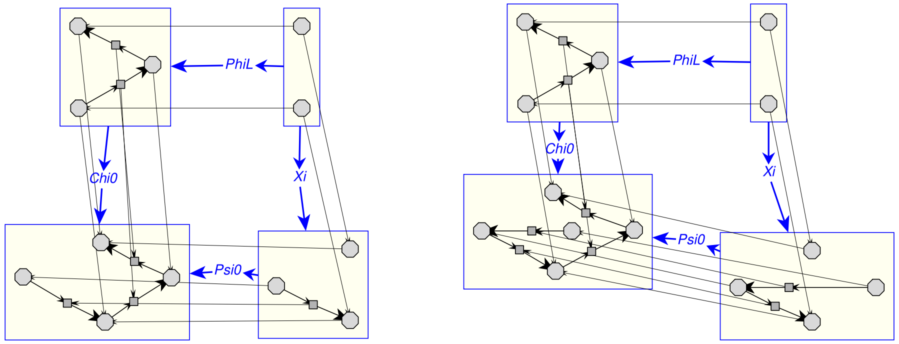
Another class of problems is illustrated by the following matching:




Intuitively, there seem to be two choices for the host:




But in both cases, the pushout of the host morphism and the left-hand-side morphism fails to reconstruct the original application graph:



The problem here is the fact that in the application graph, an edge is incident in a node that is outside the image of the interface graph. Therefore, in addition to the identification condition, also the following needs to be satisfied:

**Definition 1.2.2** For two graph morphisms $\Phi : \mathcal{G} \to \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \to \mathcal{A}$, the *dangling condition* holds iff whenever an edge connects a node outside the image of X with a node $x$ inside the image of X, then $x$ has in its pre-image via X only nodes in the image of $\Phi$.□

It happens that these two conditions together are necessary and sufficient for the existence of pushout complements, so they are usually taken to be parts of a single condition:

**Definition 1.2.3** [←107] For two graph morphisms $\Phi : \mathcal{G} \to \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \to \mathcal{A}$, the *gluing condition* holds iff both the identification condition and the dangling condition hold. □

Obviously, something is wrong with this condition.

Remember that the motto of the categoric approach is that graph rewriting should be defined entirely on an abstract level, namely the level of category theory. The rewriting step itself also is defined entirely on this level. Nevertheless, this crucial application condition is given on the concrete level of nodes and edges.

This "escape" is, in fact, not necessary: already in 1987 Kawahara gave an appropriate abstract formulation of the gluing condition [Kaw90]. For this purpose, he employed a relational calculus embedded in the more advanced category-theoretic concepts of topoi [Gol84]. However, this abstract formulation seems not to have received the deserved attention; even in the "Handbook of Graph Grammars and Computing by Graph Transformation" [Roz97], written a decade later, the exposition of the foundations of the categoric approach [CMR+97] still states the gluing condition on the concrete level of vertices and edges and does not even refer to Kawahara's abstract formulation.

In Sect. 5.4 we are going to present more details of Kawahara's formulation, and put it into a wider context.

### 1.2.5 Restricting Derivations

One can argue that in the above counter-examples to the gluing condition, application of the rules in question still makes sense, although the pushout of the rule's left-hand side and the host morphism does not reproduce the application graph.

Since there still is a commuting square, the pushout condition guarantees that there is a morphism from the pushout object to the application graph, and Parisi-Presicce developed together with Ehrig the approach of *restricting derivations* [PP93, EPP94] that allows rewriting under such circumstances.

An application of a rule $\mathcal{L} \longleftarrow \mathcal{G} \longrightarrow \mathcal{R}$ to an application graph $\mathcal{A}$ via a matching $\mathcal{L} \longrightarrow \mathcal{A}$ starts with constructing *any* host object and host morphism $\mathcal{G} \longrightarrow \mathcal{H}$ such that the matching factors via the pushout of $\mathcal{L} \longleftarrow \mathcal{G} \longrightarrow \mathcal{H}$. The result is then obtained as the pushout of $\mathcal{H} \longleftarrow \mathcal{G} \longrightarrow \mathcal{R}$, as in the double-pushout approach.

$$
\begin{tikzcd}
 & \mathcal{L} \arrow[dl] \arrow[d] & \mathcal{G} \arrow[l] \arrow[r] \arrow[d] & \mathcal{R} \arrow[d] \\
\mathcal{A} & \mathcal{A}' \arrow[l] & \mathcal{H} \arrow[l] \arrow[r] & \mathcal{B}
\end{tikzcd}
$$

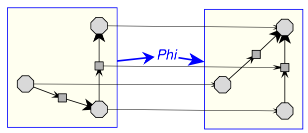


The considerable flexibility arising from this very weakly limited choice of hosts on the one hand lets restricting derivations not only encompass all double-pushout rule applications and also all single-pushout derivations (see below), but even exceed their possibilities. On the other hand, however, the number of choices in general becomes so large that this approach is not useful for automatic application without further guidance.

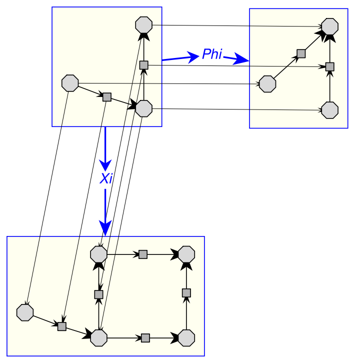
### 1.2.6 The Single-Pushout Approach

In the double-pushout approach, as we have seen above, a rule is a span $\mathcal{L} \xleftarrow{\Phi_{\mathrm{L}}} \mathcal{G} \xrightarrow{\Phi_{\mathrm{R}}} \mathcal{R}$, where $\Phi_{\mathrm{L}}$ is usually restricted to be a monomorphism, that is, an injective graph homomorphism. From the category-theoretic point of view, such a span is precisely a *partial morphism*. This can be understood by considering the monomorphism as designating a *subobject* which is then the domain of definedness of the partial morphism.

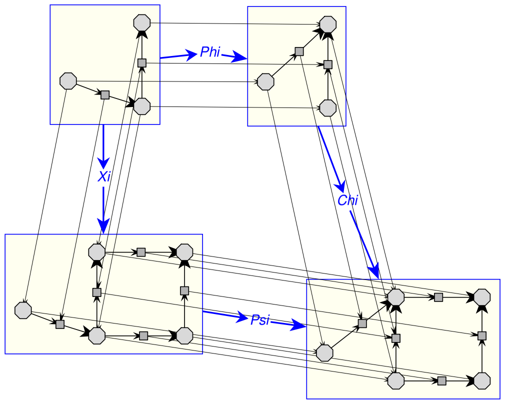
In the single-pushout approach, a rule is a single partial morphism $\mathcal{L} \xrightarrow{\Phi} \mathcal{R}$. Usually, the subobject underlying such a partial morphism is not explicitly mentioned nor drawn in this approach. The following example rule deletes an incoming edge from the source node of another edge, and replaces it by an incoming edge to the target node of that other edge:



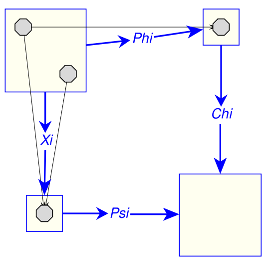

A redex is a *total* morphism from the left-hand side $\mathcal{L}$ into some application graph $\mathcal{A}$. There is general consensus that admitting partial morphisms in this rôle does not make much sense.

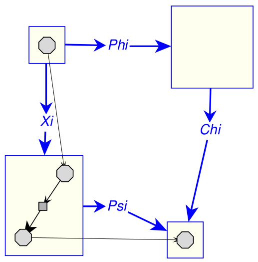



A rewriting step is now a single pushout *in the category of partial morphisms.*




Partialities in the rule morphism may be interpreted as "prescriptions to delete", while defined parts of the rule morphism may be interpreted as "prescriptions to preserve". With these interpretations, conflicts can arise: Below, in the example on the left a node that is to be deleted and a node that is to be preserved are mapped to the same node in the application graph. It is easy to see that the requirement that the pushout square must commute gives priority to deletion.




In the example on the right, a deleted node is mapped to a node with a "dangling" edge. In the double-pushout approach, such a matching would violate the dangling condition, so no derivation would be possible. In the category of partial graph morphisms, however, a pushout exists, and deletes the dangling edge, too. As a result, in circumstances like those exemplified by these two cases, the morphism from the right-hand side of the rule to the result graph is not total, even though the matchings from the left-hand side to the application graph are always total.

If one is interested only in rewriting steps with total result morphisms, then the following condition has to hold for the matching (the name "conflict-free" has been introduced by Löwe; Kennaway calls this condition IDENT' [Ken91]):

**Definition 1.2.4** A total matching morphism $\Xi : \mathcal{L} \to \mathcal{A}$ is called *conflict-free* for a (partial) rule morphism $\Phi : \mathcal{L} \nrightarrow \mathcal{R}$ iff whenever a node in the domain of $\Phi$ has the same image via $\Xi$ as another node, then that other node is in the domain of $\Phi$, too. □



In the literature on the single-pushout approach, this condition is always stated like that, on the concrete level of nodes and edges. We provide an abstract variant in Sect. 5.4.

### 1.2.7 Shortcomings of the Pushout Approaches

We have seen that with pushout rules, it is possible to specify the following:

- deletion of graph items in the image of the rule's left-hand side,
- in the single-pushout approach or via restricting derivations, deletion of "dangling" items, usually edges,
- addition of graph items,
- identification of graph items.

However, it is not possible to specify proper *replication*. The case that another, equal item is inserted for a preserved matched item does not count as proper replication. With proper replication, we mean replication of parts of the graph that are only *indicated* by the matching, but not fully specified. The kind of replication that is most interesting for us allows *graph variables* to be matched to whole subgraphs, and a rewriting step may then involve producing several copies of these subgraphs. Because of the analogy of this scenario with that of term rewriting, there has been a first attempt in this direction in the shape of "graph rewriting with substitutions" [PH96], which will be discussed in Sect. 1.3.

Another kind of replication can be found in vertex replacement graph grammars, where new edges introduced by the rule are replicated as often into the result graph as there are appropriate neighbours of the redex node.

Considering again the pushout construction, the inability to perform such arbitrary replications is clearly connected with the proximity of pushouts to disjoint sums. Since replication can be seen as an instance of a product construction, it is natural to look into the dual of the pushout construction, namely the *pullback*.

### 1.2.8 The Pullback Approach

Bauderon proposed a different categorical setting for graph rewriting that overcomes the lack of replication capabilities in the pushout approaches. Starting from the fact that the most natural replication mechanism in category theory is the categorical product, and that, in graph-structure categories, pullbacks are subobjects of products, Bauderon introduced a setup that uses pullbacks in place of pushouts [Bau97, Bau95, BJ96, BJ97, Jac99, BJ01]. He showed that it is possible to encode the rules of important vertex replacement graph grammars as single- resp. double-pullback rules — these vertex replacement systems are beyond the reach of the pushout approaches because of the unpredictable replication of edges incident to the redex node. In addition, single- and double-pushout rules can be encoded as appropriate pullback rules, too. Therefore, from the point of view of expressiveness, the pullback approach is clearly more powerful than the pushout approaches.

However, the pullback approach also has serious drawbacks.

From the technical point of view, the existence condition for pullback complements is much more complicated than the gluing condition, and it is, in general, intractable to check by computer. Therefore, at least the double-pullback approach is not appropriate as a model for implementations of graph rewriting.

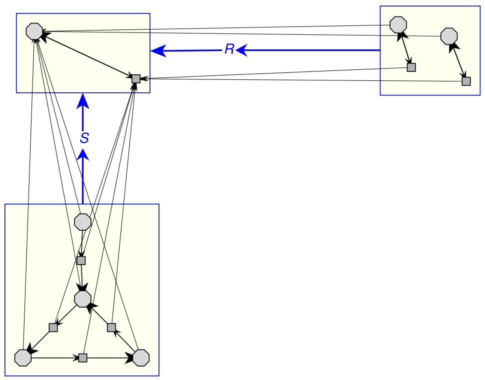
Furthermore, the gain in expressiveness comes at the cost of a severe loss in comprehensibility and intuitiveness.

Let us consider a few example pullbacks. Starting from two morphisms $R$ and $S$ with common target $\mathcal{D}$, a pullback consists of a *pullback object* $\mathcal{A}$ and two morphisms $P$ and $Q$ starting from $\mathcal{A}$ such that the resulting square commutes, and such that each commuting candidate square can be uniquely factorised via $\mathcal{A}$ (see Def. 5.1.1 for the precise definition).

$$\begin{tikzcd}
\mathcal{D} & \mathcal{B} \arrow[l, "R"'] & \\
\mathcal{C} \arrow[u, "S"] & \mathcal{A} \arrow[l, "Q"'] \arrow[u, "P"'] & \\
 & & \mathcal{A}' \arrow[ull, "Q'"] \arrow[ul, dotted, "U"'] \arrow[uul, "P'"']
\end{tikzcd}$$

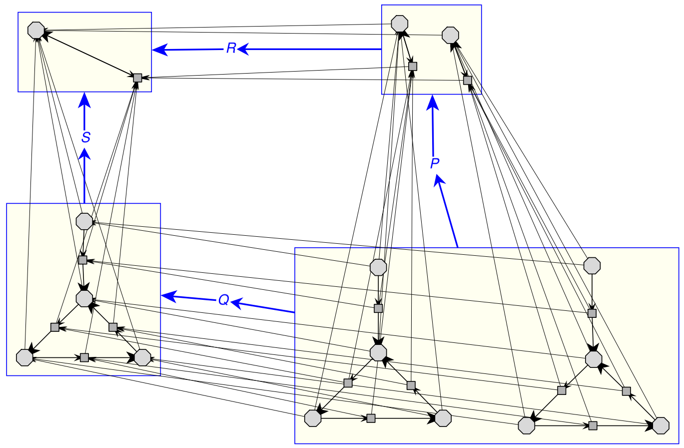
Already the "simplest special case" of pullbacks, namely the categorical product, is only simple to understand in special cases. The categorical product arises if $\mathcal{D}$ is a terminal object. In the category of graphs, a terminal object is a graph consisting of a single node with a single loop edge attached to it; such a graph is also called *unit graph*.

In the following diagram of functional graph morphisms, the common target $\mathcal{A}$ is such a unit graph, and the graph $\mathcal{B}$ is a direct sum of the unit graph with itself.



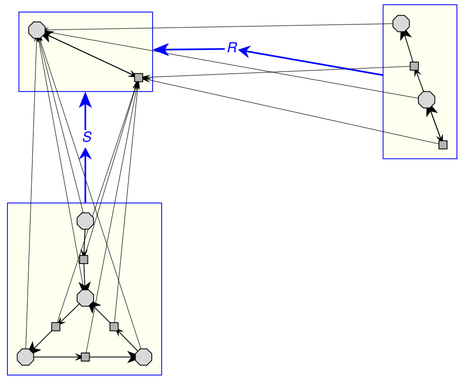

The categorical product of $\mathcal{B}$ with $\mathcal{C}$ is therefore the direct sum of $\mathcal{C}$ with itself:





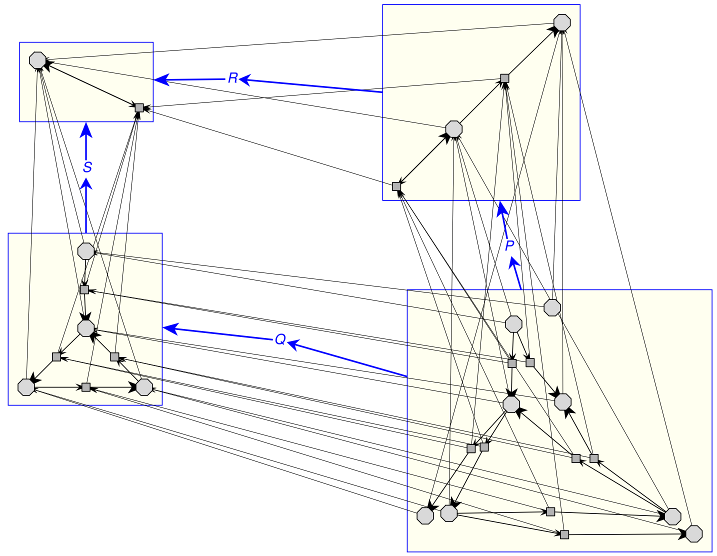

This kind of replication is easy to comprehend, and is exactly what is missing from the pushout approaches.

However, even slightly more general products are less understandable; consider the following example setup:



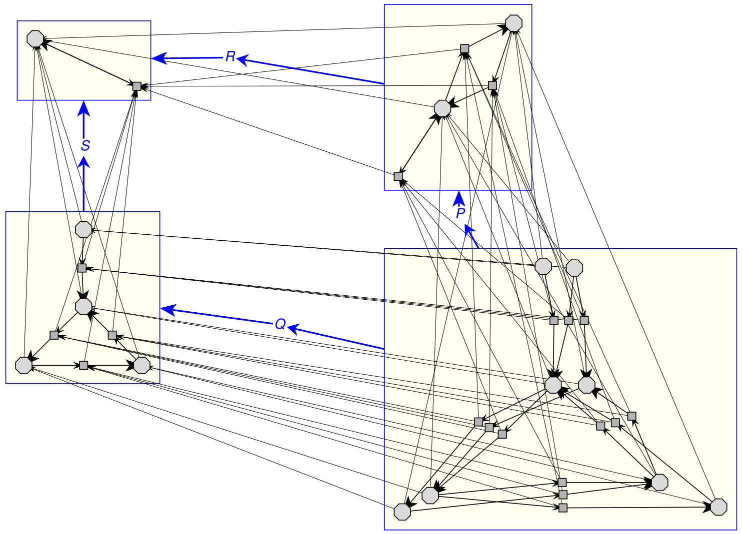

Here, the $\mathcal{B}$ still contains a unit subgraph, but it also contains an edge from the node of the unit subgraph to another node.

The direct product still includes a full copy of $\mathcal{C}$, induced by the unit subgraph. All vertices of $\mathcal{C}$ also have a second copy, corresponding to the second node of $\mathcal{B}$, and this second copy of node $x$ has an edge from the unit-image copy of node $y$ iff there is an edge from $y$ to $x$ in $\mathcal{C}$.




Edges in the opposite direction are not produced since there is no such edge in $\mathcal{B}$.

If we include such a third edge into $\mathcal{B}$, then all edges of $\mathcal{C}$ are triplicated, while the nodes are still duplicated because of the two nodes in $\mathcal{B}$.

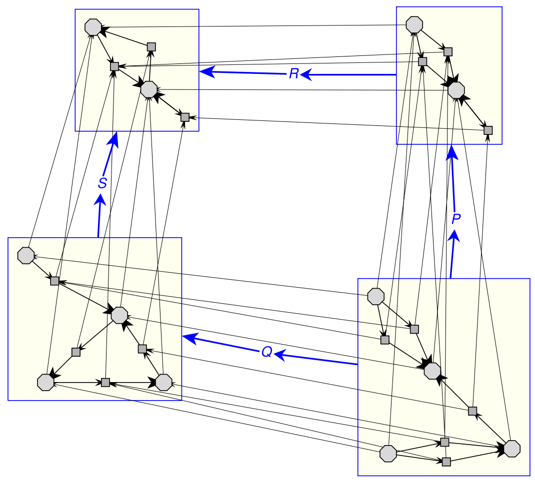



In general, the product construction involved here is known as the *Kronecker product* of graphs [FW77]: The Kronecker product of $\mathcal{B}$ and $\mathcal{C}$ contains a node $(b, c)$ for every pair of nodes $b$ of $\mathcal{B}$ and $c$ of $\mathcal{C}$, and for every pair of edges $e : b_1 \to b_2$ in $\mathcal{B}$ and $f : c_1 \to c_2$ in $\mathcal{C}$ it contains an edge from $(e, f) : (b_1, c_1) \to (b_2, c_2)$.



The intricate interlacing of the different aspects of the different copies we have seen in the examples can be reconstructed and justified from this definition of the categorical product of graphs, but can hardly be called intuitive.

Let us now show an example for a pullback which is not just a categorical product. From the point of view of $\mathcal{C}$, the part which $S$ maps into the unit subgraph of $\mathcal{D}$ is preserved, since $R$ has only a unit subgraph as pre-image of the unit subgraph in $\mathcal{D}$. The edge mapped to an edge outside the range of $R$ is deleted, while those edges mapped to an edge that has two pre-images via $R$ are duplicated:



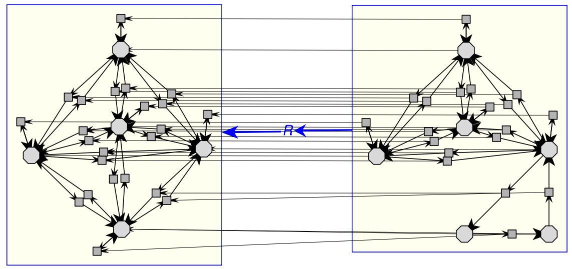

This shows that different parts of $\mathcal{D}$ can be considered as standing for different "rôles" in the rewriting step. The rule morphism $R$ decides the behaviour of each rôle, while the morphism $S$ from the application graph into $\mathcal{D}$ assigns rôles to all parts of the application graph. For maximum flexibility in the treatment of these rôles, Bauderon and Jacquet therefore provide an *alphabet graph* that contains items that each represent a different treatment via rewriting, such as preservation, duplication, or deletion.

The following shows the translation of a simple NLC rule ("node-label-controlled graph rewriting", one variant of vertex replacement), where $\mathcal{D}$ is the alphabet graph for three labels, which are represented by the horizontally flattened three-node clique in the middle. The top node is the "context", and the bottom node is the "redex". The rule morphism $R$ rewrites the redex by splitting it into two nodes connected by a single edge, redirecting incoming edges from one of the three kinds of neighbours to the source of that new edge, and redirecting the source tentacles of outgoing edges directed at the same kind of neighbours to the target of the new edge.

Context:

Labels:

Redex:




Such a rule is applied by establishing what Bauderon and Jacquet call a “label” on an application graph, that is, a graph morphism from the application graph to the alphabet graph such that exactly one node is mapped to the “redex” node, precisely its neighbours are mapped to label nodes in the alphabet graph (which are the neighbours of the redex node), and all other nodes are mapped to the “context” node.

In the following example, the bottom node is mapped to the redex; its upper three neighbours to the right-most label, and its left neighbour to the left-most label. The two top nodes are mapped to the context node, and for every edge, its mapping is determined by the mapping of its end nodes:

Context:

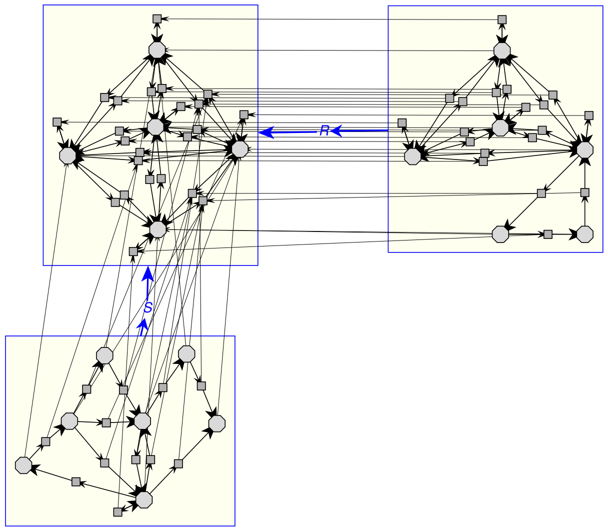
Labels:

Redex:

Context:

Neighbours:

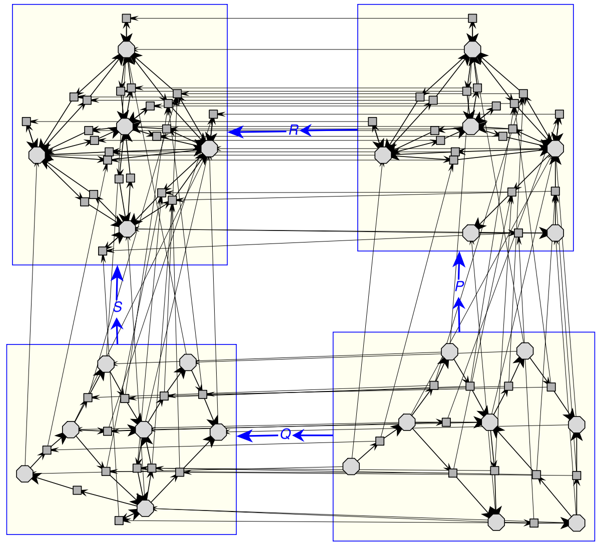
Redex:




The pullback then can be understood to remove the redex node together with all edges connecting it with its neighbours, to introduce a copy of the redex image (the original vertex replacement rule’s right-hand side) and connect it with edges to and from the neighbours of the redex, according to the prescriptions in the rule.

In our example, for each of the two edges connecting the redex image with its neighbours there are two instances in the result, connected with different neighbours according to the situation in the application graph.






This simple example should be sufficient to demonstrate that pullback rewriting “has too much power for its own good”: Although pullback rules can be used to encode many different kinds of graph rewriting rules, they are not intuitive enough to be considered as a *specification language* for graph transformation.

## 1.3 Graph Transformation with Relational Matching

Rewriting in the presence of relational morphisms is extremely interesting and important, since it offers a much higher degree of flexibility than rewriting with functional morphisms. To date, rewriting with effects like those of relational matchings has usually been achieved either via introducing some kind of substitutions, as in the purely substitution-based approach by Plump and Habel [PH96], or via intermediate steps that impose hierarchical structure, allowing functions to match “variables” to targets at higher levels in the hierarchy, as for example the following two approaches.

Drewes, Hoffmann and Plump extend the double-pushout approach to “hierarchical graphs” where hyperedges called “frames” may “contain” a nested hierarchical graph as contents [DHP00a, DHP00b]. Motivated by programming analogies, they propose “frame variables” to allow arbitrary deletion and copying of frame contents, and solve the associated problems *outside* the algebraic approach, by considering “rule schemata”: instead of allowing frame variables to match arbitrary frames, they first instantiate the rule schema to produce a rule without frame variables, and then this rule is applied. Accordingly, when discussing induced “flattened” transformation steps, they only consider rules, but not rule schemata.

The present author proposed a similar mechanism for relational diagrams, where edges are labelled with relation-algebraic expressions, and diagrams with designated interface nodes are considered as a special kind of relation-algebraic expressions [Kah99]. Although that approach went further towards directly considering relational matchings between (flat or hierarchical) diagrams, it also relied on the hierarchical view to perform the transformations themselves.

The root of the problem here seems to lie in an inappropriate concept of “variable” or “parameters”. All these approaches essentially propose (hyper-)edge variables, and then would allow these hyperedges to be matched to larger subgraphs, corresponding to the hyperedge replacements of hyperedge replacement graph grammars.

However, such a view does not give rise to a natural graph morphism concept. Although categories can be defined where morphisms contain “hyperedge substitution” components, these do not generalise naturally to truly relational morphisms.

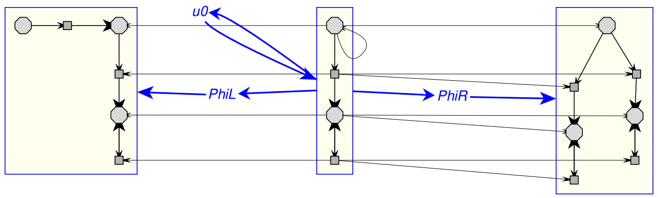
Consider the pullback in the last drawing above as a diagram with “all arrows turned around”: It is then a diagram of *relational graph homomorphisms*. Such relational graph homomorphisms are going to be defined in Def. 4.3.3 (see also Def. 3.2.2 and the discussion preceeding it); they obey similar structure preservation constraints as standard, functional homomorphisms, but can be *partial*, and *multivalent*.

In this new view, the converse of the morphism $R$ maps the redex unit graph to the subgraph that is going to replace it, a subgraph containing two vertices connected by a left-to-right edge. Note that although the pullback encodes a *vertex* replacement rule, the redex is a full unit subgraph, consisting of a vertex *and an edge*, and this edge is necessary for relating the redex with the *whole* subgraph that is going to replace it.

Analogously, approaches that extend edge replacement to relational matching cannot give rise to natural formulations as long as there are no vertices accompanying the variable edges and “covering” the vertices in the replacement graph.

In Chapter 6 we are going to present a formalism that allows the abstract specification of graph transformation with parameters and replication of parameter images. It will superficially be modelled at the double-pushout approach, but works in the setting of relational graph morphisms, and it incorporates elements of the pullback approach along the lines of the above argument.

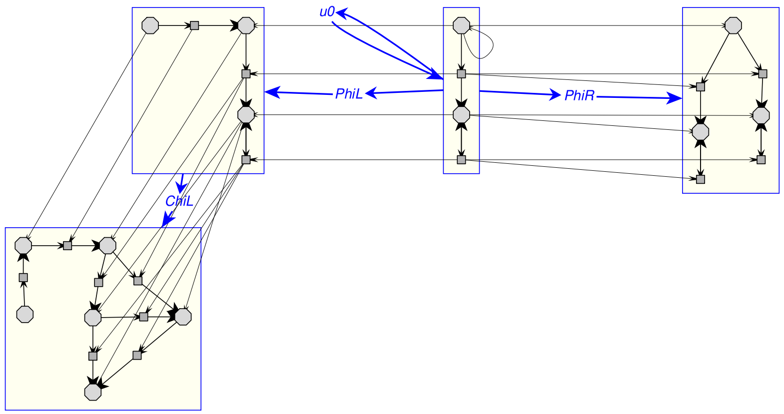
The following is an example rule:



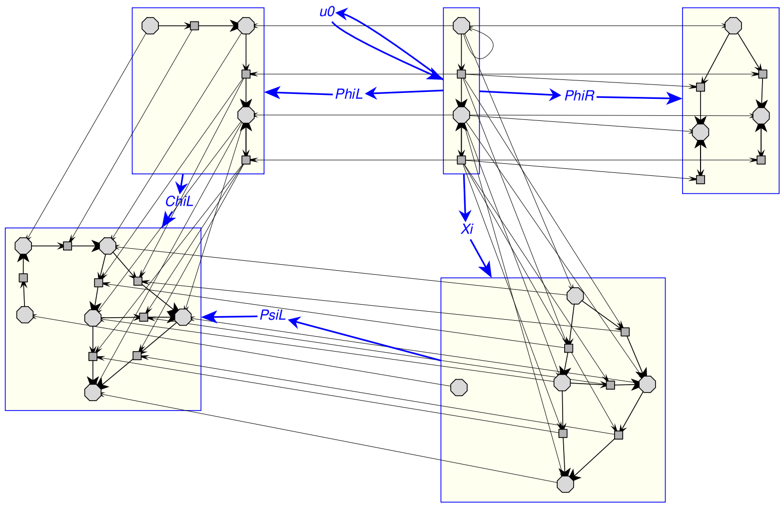

This rule consists of left- and right-hand side morphisms as usual, and has an additional component: a partial identity $u_0$ on the gluing graph that indicates the *interface*. Here, the interface consists only of the top-most node. Everything besides the interface is considered as *parameter*, and we see that the right-hand side of the example rule duplicates the parameter part.

A redex is a relational graph morphism, too, and is restricted to be total and univalent besides the parameter part. (For simplicity, we leave further more technical conditions to the detailed presentation in Chapter 6.) It may map the parameter part to a larger subgraph, as in the following example:

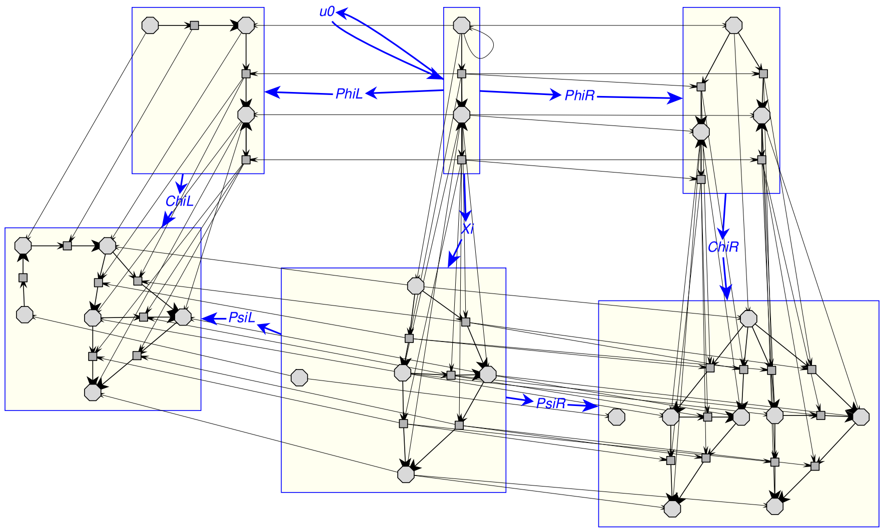





The host construction is essentially the same as in the double-pushout approach (but we are dropping a dangling edge here):




For the result, the parameter part is replicated according to the prescription of the rule’s right-hand side:




We are able to specify this kind of rewriting with fully abstract definitions that never need to mention edges or nodes. This is made possible by working in a relation-algebraic setting, which, we claim, is the right level of abstraction for graph-structure transformation.

Let us now briefly summarise the development and characteristics of relation-algebraic formalisms.

## 1.4 Relation Algebras and Generalisations

Just as lattices are a convenient abstraction from the properties of subset and substructure orderings, categories are a popular abstraction from the properties of functions between sets.

Category theory is also the abstraction underlying the so-called "algebraic approach" to graph transformation, including the slightly more abstract approach of "high-level replacement systems" [EHKPP91].

Since we are aiming at an approach to graph rewriting that allows *relational* matching, we need a more appropriate class of structures.

Attempts at characterising the behaviour of relations between sets on a more abstract level can be traced back to the roots of the "algebra of logic" in the second half of the 19th century, with landmark contributions by George Boole [Boo47], Augustus De Morgan [DM60], Charles Sanders Peirce [Pei70], and Ernst Schröder [Sch95].

In its modern shape, the study of abstractions on the behaviour of relations goes back to Alfred Tarski [Tar41, Tar52]. However, *relation algebras* as axiomatised by Tarski are *homogeneous*, that is, they abstract from the notion of relations between objects of a *single universe*.

Approaches to axiomatise relations between elements of *different sets*, that is, *heterogeneous* relation algebras, seem to have emerged only during the 1970's and 1980's, approaching the topic from two different directions.



One direction directly generalises relation algebras to the heterogeneous setting, and has Gunther Schmidt as main proponent [Sch76, Sch77, Sch81a, SS85]. One advantage of this approach is that Gunther Schmidt and Thomas Ströhlein have written a very accessible textbook [SS89, SS93] which is based on this definition of heterogeneous relation algebras and aimed at computer scientists; a more recent, more compact account may be found in a survey book chapter by Schmidt et. al. [SHW97], with applications in other chapters of the same book [BKS97].

The other direction starts from the heterogeneous setting of category theory and adds additional structure and axioms in order to be able to approach the elegance of relational reasoning.

One of the earliest steps in this directions seems to have been by Kawahara [Kaw73c, Kaw73a, Kaw73b], who established a *relational calculus* inside topos theory[2]. Unfortunately, this work seems to have received little attention.

Olivier and Serrato [OS80] introduced "Dedekind categories" as a variant of relation algebras without complementation, and proposed to use them for modelling "fuzzy relations" in [OS82].

The "$*$-autonomous categories" of Michael Barr [Bar79] and the "cartesian bicategories" of Carboni and Walters [CW87] are two, albeit quite different, attempts to harness rather abstract category-theoretic machinery in order to gain certain approximations to relational reasoning.

The current standard terminology for weaker variants of "relation categories" has been established by the book "Categories, Allegories" by Peter Freyd and Andre Scedrov [FS90]. As a first approximation to relational reasoning they define *allegories* as a specialised kind of categories. They then proceed to introduce a hierarchy of stronger structures, like "locally complete unitary pretabular allegories" (LCUPAs); their main motivations are representation theorems and to establish links with topos theory (i.e., with categorical logic). Unfortunately, the very concise and abstract style of this book together with the fact that it extensively uses notation and terminology that are not very wide-spread make it rather inaccessible for most computer scientists who are not already quite versed in category theory.

Backhouse and his group made extensive use of relators for their "Relational Theory of Datatypes" [ABH+92]. Closely related is the "Algebra of Programming" as presented in the book by Bird and de Moor [BdM97]; it extensively uses relational calculus for program derivation. Bird and de Moor tend to avoid complementation of relations, but do not definitely exclude it. So the preferred setting of the "Algebra of Programming" is essentially that of: "complete tabular power allegories", which can be regarded as a topos setting, and also corresponds to Dedekind categories with unit, direct sums and products, subobjects, and power objects.

Since the early 1990s, the "RelMiCS" initiative[3] initiated by Gunther Schmidt brings together researchers from many different approaches to using relational methods in computer science, and has also documented the foundations and their wide-ranging applications in a book [BKS97] and an on-going series of conference proceedings [HF98, JS99, Orl98, Des00].

[2] A topos is a category equipped with structure that allows to translate logical reasoning into the categorical language. Standard references include [Gol84, BW84].

[3] "Relational Methods in Computer Science", URL: http://ist.unibw-muenchen.de/relmics/

## 1.5 Contrasting the Relational and the Categoric Approaches

Let us use the notion of *direct product* to illustrate the different characteristics of categoric and relational characterisations.

In category theory, a (categorical) *product* of two objects $\mathcal{A}$ and $\mathcal{B}$ in a category is a triple $(\mathcal{P}, \pi, \rho)$ consisting of an object $\mathcal{P}$ and two *projections* $\pi : \mathcal{P} \to \mathcal{A}$ and $\rho : \mathcal{P} \to \mathcal{B}$, such that

- for all objects $\mathcal{C}$, and
- for all morphisms $f : \mathcal{C} \to \mathcal{A}$ and $g : \mathcal{C} \to \mathcal{B}$,
- there is exactly one morphism $h : \mathcal{C} \to \mathcal{P}$,
- such that $f = h;\pi$ and $g = h;\rho$.

$$\begin{tikzcd} & \mathcal{C} \arrow[dl, "f"'] \arrow[d, dotted, "!h"] \arrow[dr, "g"] & \\ \mathcal{A} & \mathcal{P} \arrow[l, "\pi"] \arrow[r, "\rho"'] & \mathcal{B} \end{tikzcd}$$

This so-called *universal characterisation* is a "global" condition — for testing whether a triple is a direct product, one needs to check for *all* objects of the category the *complete* homsets between these objects as sources and $\mathcal{A}$ and $\mathcal{B}$ as targets.

In heterogeneous relation algebras (and appropriate allegories), a direct product of two objects $\mathcal{A}$ and $\mathcal{B}$ is a triple $(\mathcal{P}, \pi, \rho)$ consisting of an object $\mathcal{P}$ and two *projection relations* $\pi : \mathcal{P} \leftrightarrow \mathcal{A}$ and $\rho : \mathcal{P} \leftrightarrow \mathcal{B}$ for which the following conditions hold [Sch77, SS93] (we use $\mathbb{I}$ to denote identity relations, $\mathbb{T}$ for universal relations, $\sqcap$ for the meet (intersection) of relations, and $\breve{}$ for converse):

$$\pi\breve{};\pi = \mathbb{I} \ , \qquad \rho\breve{};\rho = \mathbb{I} \ , \qquad \pi\breve{};\rho = \mathbb{T} \ , \qquad \pi;\pi\breve{} \sqcap \rho;\rho\breve{} = \mathbb{I} \ .$$

This is a perfectly "local" condition, which involves only the projection morphisms and primitive relations, and no quantifications at all.

In addition, the relational characterisation is syntactically first-order, in that it only involves equations, while the categorical characterisation is second-order, involving variable bindings introduced by the quantifications. Therefore, relational arguments are much more accessible both for human readers and to mechanised proof checking.

The way these semantically closely connected definitions differ in the two approaches is in fact typical for the different characteristics of categorical and relational reasoning.

Besides the sheer power of graph rewriting with relational matching, these different characteristics are another motivation for our relational approach to graph rewriting.



## 1.6 Overview

Chapter 2 first of all introduces graph structures as a special case of many-sorted algebras. Basic facts about subalgebra lattices are established, and it is shown how substructure lattices of graph structures are especially well-behaved. This allows to reason about parts of graphs (or graph structures) and many aspects of relations between such parts purely on the lattice-theoretic level. This is a useful introduction to abstract reasoning about graphs, and also serves as a foundation that will be useful in later chapters.

The next two chapters show how graphs fit into relation-algebraic formalisms. Chapter 3 establishes that relational morphisms between many-sorted algebras in general already obey the laws of allegories — one of the simplest variants of relation-algebraic formalisms. For graph structures, many more laws hold. As is shown in Chapter 4, every graph-structure signature gives rise to a strict Dedekind category of graph structures with relational morphisms. This means that apart from laws that require the existence of complements, all the laws of heterogeneous relation algebras hold.

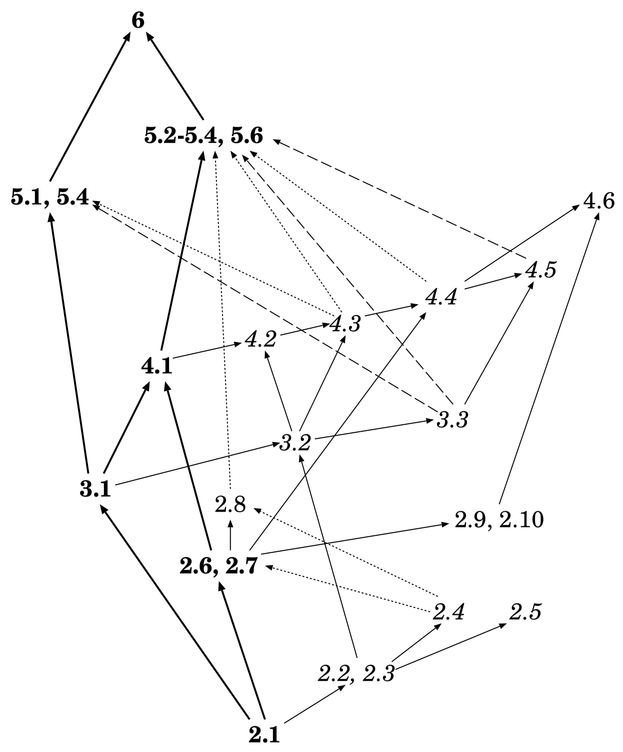
The focus of these Chapters 2–4 is mainly to establish the lattice, allegory, and Dedekind category laws for the respective classes of algebras on an elementary level, and to promote an intuitive understanding how relational reasoning carries over to the setting of graph structures. The theory mostly follows from established category-theoretic facts, namely that every graph-structure category is a topos. However, we feel that using topos theory for explaining relational graph morphisms and also graph transformation is overkill, and the "natural" language of topos theory is in fact not well-suited for this task. Therefore, we do not expose the reader to topos theory at all, and rather chose to present the relation-algebraic approach to relational graph morphisms in a self-contained, more textbook-like manner.

In Chapter 5 we show how the arsenal of the categoric approaches to graph transformation can be recovered with relation-algebraic means. For the double-pushout approach, we may build on the results of Kawahara, who used a relational calculus embedded in topos theory to formulate the gluing condition; for the single-pushout approach and for pullback complements we are not aware of any previous results in this direction. Already this material should amply demonstrate that the language of relations is perfectly suited for describing graph transformations in a fully abstract way, without ever having to resort to concepts defined on the concrete level of nodes and edges, like for example the traditional presentations of the gluing condition.

Finally, in Chapter 6 we show how the relational context makes it straightforward to define rewriting concepts that integrate the intuitive understandability of the double-pushout approach with the replicative power of the pullback approach. We first show a straightforward amalgamation of the two approaches, and then continue to modify this concept towards allowing more general relational matchings and rule morphisms.

Some auxiliary properties of allegories and Dedekind categories have been collected in Appendix A, while Appendix B contains the lengthier proofs for Chapter 6.

The following graph sketches the essential dependencies between different parts of this thesis. The bold part on the left is the main development of a fully component-free relation-algebraic approach to rewriting. The sections in italics provide concrete and abstract algebras and graph structures that can serve as instantiations to the abstract approach, and that serve as motivational examples and illustrations, as documented via the dotted arrows.




The three dashed arrows refer to the abstract relation-algebraic definitions of subobjects, quotients and direct sums — these definitions are presented in the context of their algebraic instantiations instead of inside sections 3.1 and 4.1, respectively.

The current investigation is aimed at computer scientists with an interest in graph transformation and with a standard background in discrete mathematics, including some relational calculus. Some passages are intended to satisfy the needs of readers with a stronger mathematical or category-theoretic background. Therefore, we strive to give self-contained definitions before using advanced concepts, and state important properties; but for facts that can be found in the standard literature, we only provide general links into that literature and omit proofs and detailed attributions.



## 1.7 Eddi

The graph diagram drawings in Sect. 1.2 and similar drawings throughout this thesis have been generated using "Eddi", a prototype editor for directed graph-structure diagrams, that also supports relational graph-structure transformation.

It uses the library RATH [KS00], which provides datatypes and a framework for categories, allegories and relation algebras. This framework has been instantiated for Dedekind categories of graph structures over unary signatures. A graphical front end allows to draw diagrams of graph structures with relational morphisms between them in a WYSIWYG manner, and to invoke relational operations including all the graph transformation constructions presented in this thesis.

More information about this tool is available from the Eddi home page:

URL: http://www.cas.mcmaster.ca/~kahl/Eddi/

## Chapter 2

# Graph Structures and Their Parts

Graphs are usually presented as tuples consisting of node and edge sets, and of two total functions from edges to nodes, assigning every edge a source and a target node. Such a tuple can equivalently be considered as a (two-sorted) *algebra* over a particular signature.

In computing, we are used to consider algebras over fixed signatures as implementations of data types with a fixed interface. Data are then elements of the carriers of these algebras. This view is the motivation of most of the study of algebras in computer science, see e.g. [EM85].

However, there are also approaches that consider algebras themselves as *data*. A rather general approach in this direction is that of *abstract state machines*, formerly called "evolving algebras", by [Gur91, GKOT00].

A more specialised instance is computing using graph-like structures. There are many different graph-like structures, that all can be considered as algebras over appropriate signatures. When looking for a precise definition what "graph-like" actually means, it is useful to scrutinise the different signatures underlying "graph-like" structures. It then turns out that one useful characteristic is that many of these signatures are *unary*, i.e., function symbols have only one argument. If we use this as characterisation, then graph-like structures are *unary algebras*. Many well-known graph-like structures fall into this class, among them standard graphs, and many different kinds of hypergraphs.

A category-theoretic approach to algebras was initiated by Lawvere in a pioneering paper [Law63], and although we present concrete algebras in this chapter, we essentially reflect Lawvere's approach by generalising concrete algebras without real technical effort to abstract algebras in Def. 3.2.1 in the next chapter.

In this chapter we study properties concerning substructures of graph-like structures. Some properties are about individual substructures; others concern the relation between substructures.

However, since graph-like structures can be algebras over many different signatures, we use "nodes" and "edges" only in motivating discussions. For our definitions, we completely rely on the lattice properties of the lattices of subalgebras of $\Sigma$-algebras — these lattices have particularly useful properties in the case of unary signatures. We employ these properties to define concepts like discreteness and borders of subgraphs, all only using lattice operations.

We start out by fixing some notation and notions for sets and lattices. Then we introduce signatures and algebras, and present general facts about subalgebra lattices. These are then specialised in Sect. 2.4 to the case of unary signatures, which define graph-like structures. A short detour introduces partial algebras together with a particular concept of partial subalgebra, and we show that the resulting subalgebra lattices are isomorphic to those induced by a translation into unary algebras.

Pseudo-complements are well-established in lattice theory and in particular in its applications to topology; they seem, however, not to have been intensively studied in the context of graphs and subgraphs, let alone that of general subalgebras. For this reason we





present a definition of pseudo-complements that is slightly more general than usual, and show a few basic properties in Sect. 2.6. An important fact is that pseudo-complements can be used to formalise an abstract version of *induced subgraphs*.

In the discussion of subgraphs and their mutual relations, of similar importance as pseudo-complements are their duals, which we call *semi-complements*.

Using semi-complements, in Sect. 2.8 we sketch how the lattice-theoretic approach to subgraphs may already be used to produce a simple graph rewriting formalism. The — hardly surprising — shortcomings of that approach serve as an additional motivation for subsequent chapters.

Finally, we show how a whole range of (sub)graph properties, for example discreteness, separable parts, and connectedness, can be defined in a component-free way in our lattice-theoretic setting, based mostly on the definition and properties of semi-complements.

## 2.1 Preliminaries: Sets, Lattices

To a certain extent, our notation is oriented at that of Z [Spi89], a mathematical notation based on a typed set theory (and designed for specification purposes). However, no previous knowledge of Z is required; we introduce all notation we use.

### Sets

If $\mathcal{A}$ is a set, then we introduce a meta-variable $x$ for elements of $\mathcal{A}$ by writing the *declaration* "$x : \mathcal{A}$". Slightly more informally, this may be understood as introducing $x$ as an element of $\mathcal{A}$. Multiple declarations are separated by semicolons, as in "$x : \mathcal{A};\ y : \mathcal{B}$".

The notation "$x \in \mathcal{A}$" is understood as a formula, i.e., it is reserved for the statement that $x$, which should already be introduced at that point (usually as an element of perhaps some other set) is actually an element of $\mathcal{A}$.

For set comprehensions, we use the Z pattern of "$\{$*declaration* $\mid$ *predicate* $\bullet$ *term*$\}$". Here, "*predicate*" is a formula and "*term*" is an expression; both usually contain variables introduced in the "*declaration*". Such a set contains all those values of "*term*" that result from variable assignments induced by "*declaration*" for which the "*predicate*" holds. For example,

$$\{x : \mathbb{N} \mid x < 4 \bullet x^2\} = \{0, 1, 4, 9\}\ .$$

The shape "$\{$*declaration* $\bullet$ *term*$\}$" is equivalent to "$\{$*declaration* $\mid 0 = 0 \bullet$ *term*$\}$", and the shape "$\{$*declaration* $\mid$ *predicate*$\}$" should be understood as having as "*term*" the tuple of all variables in the declaration, in the order of the declaration, for example:

$$\{y, x : \mathbb{N} \mid x < y < 4 - x\} = \{y, x : \mathbb{N} \mid x < y < 4 - x \bullet (y, x)\} = \{(1,0), (2,0), (3,0), (2,1)\}$$

The empty set is denoted "$\emptyset$".

A *relation* from a set $\mathcal{A}$ to another set $\mathcal{B}$ is a subset of the Cartesian product of $\mathcal{A}$ and $\mathcal{B}$, that is, $R$ is a relation from $\mathcal{A}$ to $\mathcal{B}$ iff $R \subseteq \mathcal{A} \times \mathcal{B}$, or, equivalently, $R \in \mathbb{P}(\mathcal{A} \times \mathcal{B})$, where $\mathbb{P}(\mathcal{X})$ denotes the powerset of set $\mathcal{X}$. The Z notation introduces the abbreviation $\mathcal{A} \leftrightarrow \mathcal{B}$ for $\mathbb{P}(\mathcal{A} \times \mathcal{B})$; so we may introduce relations by writing $R : \mathcal{A} \leftrightarrow \mathcal{B}$.

From the next chapter on, we shall use more abstract concepts of relations and functions, but in the present chapter we stick with this concrete view.

In particular, a *function* (or *partial function*) is a *univalent* relation, where $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is univalent iff

$$\forall a : \mathcal{A}; b_1, b_2 : \mathcal{B} \mid \{(a, b_1), (a, b_2)\} \subseteq R \bullet b_1 = b_2 \ .$$

(Quantified formulae follow the same pattern as set comprehensions, just with a body formula instead of a body term.)

The set of partial functions from $\mathcal{A}$ to $\mathcal{B}$ is abbreviated $\mathcal{A} \nrightarrow \mathcal{B}$, and that of total functions, also called *mappings*, $\mathcal{A} \rightarrow \mathcal{B}$ (a relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is total iff for every element $a : \mathcal{A}$ there is a $b : \mathcal{B}$ such that $(a, b) \in R$).

So elements of functions are pairs — it is a wide-spread convention to use special notation for pairs in this context: "$a \mapsto b$" is considered as equivalent to "$(a, b)$".

A *family* of mathematical objects $X_i$ indexed over indices $i$ taken from some index set $\mathcal{I}$ is written "$(X_i)_{i:\mathcal{I}}$" — mathematically, this is just a function $X$ that might more correctly be defined via either a set comprehension or a $\lambda$-abstraction:

$$X := \{i : \mathcal{I} \bullet i \mapsto X_i\} = \lambda\, i : \mathcal{I} \bullet X_i$$

The "family" notion and notation are however so widespread in the mathematical literature that we also apply them where usual.

The set of finite sequences of elements from $\mathcal{A}$ is denoted $\mathcal{A}^*$. The sequence having the elements $x_1, \ldots, x_n$ in that order is written $\langle x_1, \ldots, x_n \rangle$. If $s$ is a sequence, we write $s_i$ for the $i$-th element of $s$.

### Lattices

An *ordered set* $(\mathcal{A}, \leq)$ is a set $\mathcal{A}$ together with a relation $\leq$, which has to be an *ordering relation*, i.e., it is

- reflexive: $x \leq x$ for all $x : \mathcal{A}$,
- transitive: for all $x, y, z : \mathcal{A}$, if $x \leq y$ and $y \leq z$, then $x \leq z$, and
- antisymmetric: for all $x, y : \mathcal{A}$, if $x \leq y$ and $y \leq x$, then $x = y$.

In an ordered set $(\mathcal{A}, \leq)$, an *upper bound* of a subset $\mathcal{X} \subseteq \mathcal{A}$ is an element $u : \mathcal{A}$ such that $x \leq u$ for all $x : \mathcal{X}$, and a *lower bound* of $\mathcal{X}$ is an element $l : \mathcal{A}$ such that $l \leq x$ for all $x : \mathcal{X}$.

A *greatest element* of a subset $\mathcal{X} \subseteq \mathcal{A}$ is an element of that subset $\mathcal{X}$ that is also an upper bound of $\mathcal{X}$, and a *least element* of $\mathcal{X}$ is an element of $\mathcal{X}$ that is also a lower bound of $\mathcal{X}$. Greatest and least elements are uniquely determined if they exist.

The *least upper bound* of a subset $\mathcal{X} \subseteq \mathcal{A}$ is the least element of the set of upper bounds of $\mathcal{X}$, and the *greatest lower bound* of $\mathcal{X}$ is the greatest element of the set of lower bounds of $\mathcal{X}$. If they exist, we write $\bigvee \mathcal{X}$ for the least upper bound of $\mathcal{X}$, and $\bigwedge \mathcal{X}$ for the greatest lower bound of $\mathcal{X}$.

A *lower semi-lattice* is an ordered set $(\mathcal{L}, \leqslant)$ where every finite non-empty set has a greatest lower bound. Greatest lower bounds are called *meets*, and the meet of the set $\{x, y\}$ is denoted $x \wedge y$ (existence of all binary meets is indeed sufficient for the existence of all finite non-empty meets). The binary *meet operator* $\wedge$ is



- idempotent: $x \wedge x = x$,
- commutative: $x \wedge y = y \wedge x$, and
- associative: $x \wedge (y \wedge z) = (x \wedge y) \wedge z$.

We call a lower semi-lattice *complete* if every subset $\mathcal{X}$ of $\mathcal{L}$ has a meet $\bigwedge \mathcal{X}$. The meet of the empty set, if it exists, is the greatest element of the whole semi-lattice, written $\top_{\mathcal{L}}$.

An *upper semi-lattice* is an ordered set $(\mathcal{L}, \leqslant)$ where every finite non-empty set has a least upper bound. Least upper bounds are called *joins*, and the join of $\{x, y\}$ is denoted $x \vee y$ (existence of all binary joins is indeed sufficient for the existence of all finite non-empty joins). The binary *join operator* $\vee$ is

- idempotent: $x \vee x = x$,
- commutative: $x \vee y = y \vee x$, and
- associative: $x \vee (y \vee z) = (x \vee y) \vee z$.

We call an upper semi-lattice *complete* if every subset $\mathcal{X}$ of $\mathcal{L}$ has a join $\bigvee \mathcal{X}$. The join of the empty set, if it exists, is the least element of the whole semi-lattice, written $\bot_{\mathcal{L}}$.

A *lattice* is a lower semi-lattice that is also an upper semi-lattice. In a lattice, in addition to the idempotence, commutativity, and associativity laws listed above, also the following absorption laws hold:

$$\begin{aligned} x \wedge (x \vee y) &= x \ , \\ x \vee (x \wedge y) &= x \ . \end{aligned}$$

The ordering can be regained from the binary operations:

$$x \wedge y = x \qquad \Leftrightarrow \qquad x \leq y \qquad \Leftrightarrow \qquad x \vee y = y \ .$$

The *duality principle* states that for every concept or law about all lattices, there is a *dual* concept or law which is obtained by consistently exchanging $\geqslant$ for $\leqslant$, and vice versa, and consequently also exchanging joins for meets and vice versa.

A *complete lattice* is a lattice in which all joins exist; this implies that all meets exist (and vice versa).

A *distributive lattice* is a lattice where the following (equivalent) distributive laws hold:

$$\begin{aligned} x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \\ x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \end{aligned}$$

A lattice is called *completely upwards-distributive* iff existence of the joins in the following equality is equivalent, and the equality holds where the joins exist:

$$x \wedge \bigvee \mathcal{Y} = \bigvee \{y : \mathcal{Y} \bullet x \wedge y\} \ .$$

Dually, a lattice is called *completely downwards-distributive* iff

$$x \vee \bigwedge \mathcal{Y} = \bigwedge \{y : \mathcal{Y} \bullet x \vee y\} \ .$$

A lattice is *completely distributive* iff it is both completely upwards-distributive and completely downwards-distributive.

In the literature, completely upwards-distributive complete lattices are considered under many different names, for example frames, locales, Brouwerian lattices, or Heyting algebras (for careful distinctions, see for example [Vic89]). Completely downwards-distributive lattices, however, are rarely mentioned, which may be related with the fact that the open-set lattices of topologies, which have been motivating examples for much of the research in this direction, are usually *not* completely downwards-distributive.

The concept of *algebraic lattice* will only be mentioned; an algebraic lattice is a complete lattice where every element is the join of compact elements; an element $c$ is compact iff whenever $c \leqslant \bigvee \mathcal{X}$ for some set of elements $\mathcal{X}$, then $c \leqslant \bigvee \mathcal{X}_1$ for some finite subset $\mathcal{X}_1 \subseteq \mathcal{X}$. (A similar definition defines compact elements in cpos and algebraic cpos in domain theory.)

What is more important for us than this definition is that distributive algebraic lattices are completely upwards-distributive.

A lattice element $x : \mathcal{L}$ is called *join-irreducible* iff $x = y \vee z$ implies $x = y$ or $x = z$.

In a lattice with least element $\bot$, an *atom* is an element $a : \mathcal{L}$ such that $a \neq \bot$ and for all $x : \mathcal{L}$ we have that $x < a$ implies $x = \bot$.

A lattice is called *atomic* iff for every element $x : \mathcal{L}$ with $x \neq \bot$ there is an atom $a : \mathcal{L}$ with $a \leqslant x$.

More information about lattices may be found for example in [Grä78].

## 2.2 Signatures

Signatures define the interface of abstract data types, or algebras. Signatures contain the names of the visible types of data, called "sorts", and the names of the available operations, called "function symbols" or "operators", and for these names, additional information regarding their arities, which sorts are expected to provide the arguments, and the sort of the result.

**Definition 2.2.1** [←90] A *signature* is a tuple $(\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ consisting of

- a set $\mathcal{S}$ of *sorts*,
- a set $\mathcal{F}$ of *function symbols*,
- a mapping $\mathsf{src} : \mathcal{F} \to \mathcal{S}^*$ associating with every function symbol the list of its *source sorts*, and
- a mapping $\mathsf{trg} : \mathcal{F} \to \mathcal{S}$ associating with every function symbol its *target sort*. □

For a function symbol $f : \mathcal{F}$, instead of the rather verbose "$\mathsf{src}(f) = \langle s_1, \dots, s_n \rangle$ and $\mathsf{trg}(f) = t$" we usually employ the shorthand "$f : s_1 \times \cdots \times s_n \to t$".

For a function symbol $f : \mathcal{F}$, the length of the list $\mathsf{src}(f)$ is called the *arity* of $f$.

A function symbol $c : \mathcal{F}$ with arity zero is called a *constant symbol*, and we write $c : t$ if $\mathsf{trg}(c) = t$.

A function symbol with arity one is called *unary*; one with arity two *binary*, and so on. A signature is called *unary* if all its function symbols are unary. In a context where a signature is known to be unary, we usually let $\mathsf{src}$ have just a sort as its result, instead of a singleton list of sorts.



We present signatures in a special notation; a first example for this is a (unary) signature that will be used very frequently, namely that for *graphs*; it has two sorts, $\mathcal{V}$ for *vertices* (or *nodes*), and $\mathcal{E}$ for *edges*, and two unary function symbols for source and target vertices of edges:

$$
\begin{array}{rl}
\mathsf{sigGraph} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{V}, \mathcal{E} \\
& \quad \textbf{ops: } \; s : \mathcal{E} \to \mathcal{V} \\
& \quad \qquad\quad t : \mathcal{E} \to \mathcal{V} \\
& \textbf{sig end}
\end{array}
$$

A signature that is very important for mathematicians is that for groups. It is usually considered having only one sort, and three operators of different arities: the unit as constant, inverse as unary operator, and multiplication as binary operator.

$$
\begin{array}{rl}
\mathsf{sigGroup} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{G} \\
& \quad \textbf{ops: } \; e : \mathcal{G} \\
& \quad \qquad\quad inv : \mathcal{G} \to \mathcal{G} \\
& \quad \qquad\quad * : \mathcal{G} \times \mathcal{G} \to \mathcal{G} \\
& \textbf{sig end}
\end{array}
$$

For lattices, we present two different signatures: one with only one sort, and the usual binary operations, and another one with an additional sort for sets of elements, and join and meet as unary operations on these sets:

$$
\begin{array}{rl}
\mathsf{sigLat} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{L} \\
& \quad \textbf{ops: } \; \wedge : \mathcal{L} \times \mathcal{L} \to \mathcal{L} \\
& \quad \qquad\quad \vee : \mathcal{L} \times \mathcal{L} \to \mathcal{L} \\
& \textbf{sig end}
\end{array}
\qquad
\begin{array}{rl}
\mathsf{sigCLat} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{L}, \mathcal{S} \\
& \quad \textbf{ops: } \; \bigwedge : \mathcal{S} \to \mathcal{L} \\
& \quad \qquad\quad \bigvee : \mathcal{S} \to \mathcal{L} \\
& \quad \qquad\quad \mathsf{sing} : \mathcal{L} \to \mathcal{S} \\
& \quad \qquad\quad \cup : \mathcal{S} \times \mathcal{S} \to \mathcal{S} \\
& \textbf{sig end}
\end{array}
$$

One kind of directed hypergraphs is built upon the following signature containing separate sorts for vertices, hyperedges, source and target tentacles, and function symbols to associate with every tentacle the hyperedge it starts from, and the vertex it reaches out to:

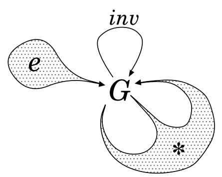
$$
\begin{array}{ll}
\mathsf{sigDHG} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathcal{T} \\
& \quad \textbf{ops: } V_S : \mathcal{S} \to \mathcal{V} \\
& \quad \phantom{\textbf{ops: }} E_S : \mathcal{S} \to \mathcal{E} \\
& \quad \phantom{\textbf{ops: }} V_T : \mathcal{T} \to \mathcal{V} \\
& \quad \phantom{\textbf{ops: }} E_T : \mathcal{T} \to E \\
& \textbf{sig end}
\end{array}
$$

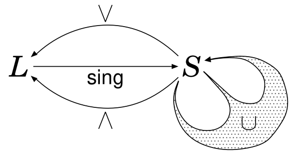
For use in examples, we introduce further signatures:

$$
\begin{array}{llll}
\mathsf{sigLoop} := & \textbf{sig begin} & \mathsf{sigB1} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{N} & & \quad \textbf{sorts: } \mathcal{N} \\
& \quad \textbf{ops: } n : \mathcal{N} \to \mathcal{N} & & \quad \textbf{ops: } f : \mathcal{N} \times \mathcal{N} \to \mathcal{N} \\
& \textbf{sig end} & & \textbf{sig end} \\
\\
\mathsf{sigTwoSets} := & \textbf{sig begin} & \mathsf{sigC1} := & \textbf{sig begin} \\
& \quad \textbf{sorts: } \mathcal{P}, \mathcal{Q} & & \quad \textbf{sorts: } \mathcal{N} \\
& \quad \textbf{ops: } & & \quad \textbf{ops: } c : \mathcal{N} \\
& \textbf{sig end} & & \textbf{sig end}
\end{array}
$$

Every signature can be seen as a directed hypergraph, where sorts are vertices and every function symbol $f : s_1 \times \cdots \times s_n \to t$ is a hyperedge, with source node sequence $s_1, \ldots, s_n$, and target node $t$.

See here hypergraph presentations of sigGroup and sigCLat:




Constants are hyperedges with no source nodes and just one target node — such hyperedges are sometimes called "loops", but since the word "loop" is also used for simple edges with the same node as source and target, we avoid using this term here.

Unary operators are conventional edges with exactly one source and one target, and therefore do not need "creative" graphical hyperedge treatment.

It is important to note that a unary signature simply is a directed graph.

## 2.3 Algebras and Subalgebras

An algebra over a signature is an *interpretation* of the syntactic material provided in that signature in appropriate semantic domains. In the simplest case, semantic objects are drawn from the realm of sets and functions between sets, and this is the standard concept of many-sorted algebras that we introduce in the following definition. However, the same mechanism of interpretation also works on more abstract domains, as we shall see in the next chapter.



**Definition 2.3.1** Given a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, a $\Sigma$-*algebra* $\mathcal{A}$ consists of the following items:

- a sort-indexed family $(s^{\mathcal{A}})_{s:\mathcal{S}}$ of *carrier sets*, i.e., for every sort $s : \mathcal{S}$, a set $s^{\mathcal{A}}$, and
- for every function symbol $f : \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$ a total function $f^{\mathcal{A}}$ from the Cartesian product $s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}$ of the source sort carriers to $t^{\mathcal{A}}$, the target sort carrier. □

We extend the interpretation of sorts to sequences of sorts: If $\mathcal{A}$ is a $\Sigma$-algebra and $s : \mathcal{S}^*$ with $s = \langle s_1, \ldots, s_n \rangle$ is a sequence of sorts, then we define

$$s^{\mathcal{A}} := s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}} \ .$$

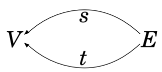
Therefore, for every function symbol $f : \mathcal{F}$, its interpretation in $\mathcal{A}$ is a total function in $(\mathsf{src}\, f)^{\mathcal{A}} \to (\mathsf{trg}\, f)^{\mathcal{A}}$.

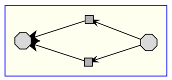
An important difference between our definition of an algebra and those usually found in text books is that we allow *empty carrier sets*. As we shall see, this is not only useful, but even necessary for obtaining a nice theoretical framework.

Most examples shall be graphs; and the $\Sigma$-algebra view on graphs is perfectly equivalent to the view that a graph is a quadruple consisting of vertex set, edge set, source mapping, and target mapping:

**Definition 2.3.2** A *graph* is a sigGraph-algebra. □

Many examples already have been seen in the introduction, so we here only show the graph underlying the signature sigGraph, once with names attached to its nodes and edges, and once as a "pure" graph:



**Definition 2.3.3** [←44] Given a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and two $\Sigma$-algebras $\mathcal{A}$ and $\mathcal{B}$, we say that $\mathcal{A}$ is a *subalgebra* of $\mathcal{B}$, written $\mathcal{A} \preccurlyeq \mathcal{B}$, iff

- for every sort $s : \mathcal{S}$, inclusion holds between the carriers: $s^{\mathcal{A}} \subseteq s^{\mathcal{B}}$, and
- for every function symbol $f : \mathcal{F}$, inclusion holds between its interpretations: $f^{\mathcal{A}} \subseteq f^{\mathcal{B}}$.

The set of all subalgebras of $\mathcal{B}$ is denoted by $\mathcal{B}_{\preccurlyeq}$. □

It is obvious that $\preccurlyeq$ is an ordering — the ordering properties are inherited from those of set inclusion $\subseteq$.

If we know that $\mathcal{A}$ is a subalgebra of $\mathcal{B}$, then we need to know only the carriers of $\mathcal{A}$ in order to be able to derive its operations — for a function symbol $f : s_1 \times \cdots \times s_n \to t$, its interpretation in $\mathcal{A}$ has to be precisely the restriction of $f^{\mathcal{B}}$ to $s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}$, since $f^{\mathcal{B}}$ is univalent and therefore this restriction is the only possibility for $f^{\mathcal{A}}$ to be total:

**Lemma 2.3.4** Given a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a $\Sigma$-Algebra $\mathcal{B}$ and a sort-indexed family $(s^{\mathcal{A}})_{s:\mathcal{S}}$ where for every $s : \mathcal{S}$ we have $s^{\mathcal{A}} \subseteq s^{\mathcal{B}}$, there is at most one $\Sigma$-algebra $\mathcal{A}$ with $(s^{\mathcal{A}})_{s:\mathcal{S}}$ as carriers. □

This allows us to specify subalgebras of a given algebra by just specifying their carriers.

It is well-known that the restriction of this ordering to the subalgebras of a given algebra $\mathcal{T}$ is an algebraic lattice ordering. The carriers of greatest lower bounds arise as intersection of the respective carriers:

**Theorem 2.3.5** Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, a $\Sigma$-algebra $\mathcal{T}$, and a subset $\mathbb{A}$ of $\mathcal{T}_{\preccurlyeq}$ be given. Then $\mathbb{A}$ has a greatest lower bound $\bigwedge_{\preccurlyeq,\mathcal{T}} \mathbb{A}$ with respect to $\preccurlyeq$ in $\mathcal{T}_{\preccurlyeq}$.

If $\mathbb{A}$ is empty, then $\mathcal{T}$ itself is the greatest lower bound of $\mathbb{A}$, otherwise, the carriers of $\bigwedge_{\preccurlyeq,\mathcal{T}} \mathbb{A}$ are $(\bigcap\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\})_{s:\mathcal{S}}$.

**Proof**: For empty $\mathbb{A}$, the statement is obvious. For non-empty $\mathbb{A}$, we define a structure $\mathcal{B}$ as follows:

- the carriers of $\mathcal{B}$ are $(\bigcap\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\})_{s:\mathcal{S}}$, and
- for every function symbol $f : \mathcal{F}$, the interpretation is the intersection of the corresponding interpretations:
$$f^{\mathcal{B}} = \bigcap\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{A}}\} \ .$$

We have to show that $\mathcal{B}$ is a well-defined $\Sigma$-Algebra: Consider a function symbol $f : \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$. If $(x_1, \ldots, x_n) \in s_1^{\mathcal{B}} \times \cdots \times s_n^{\mathcal{B}}$, then $(x_1, \ldots, x_n) \in s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}$ for all $\mathcal{A} : \mathbb{A}$. Since all $\mathcal{A} : \mathbb{A}$ are subalgebras of $\mathcal{T}$, there is a $y : t^{\mathcal{T}}$ such that $y = f^{\mathcal{T}}(x_1, \ldots, x_n)$. This implies that for all $\mathcal{A} : \mathbb{A}$ we have first $y = f^{\mathcal{A}}(x_1, \ldots, x_n)$ and therefore also $y \in t^{\mathcal{A}}$. This implies that $y \in t^{\mathcal{B}}$, so $f^{\mathcal{B}}$ is total and well-defined — note that this argument is also valid for constants, i.e., for $n = 0$. Since univalence is obvious from the definition, $\mathcal{B}$ is well-defined.

By construction, $\mathcal{B}$ is furthermore a subalgebra of every element of $\mathbb{A}$, so it is a lower bound of $\mathbb{A}$. Since its components are greatest lower bounds, this also applies to $\mathcal{B}$, which is therefore equal to $\bigwedge_{\preccurlyeq,\mathcal{T}} \mathbb{A}$. □

This theorem tells us that $(\mathcal{T}_{\preccurlyeq}, \preccurlyeq)$ is a complete lower semi-lattice. For binary meet of subalgebras $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{T}$ we write "$\mathcal{A} \wedge_{\mathcal{T}_{\preccurlyeq}} \mathcal{B}$", and where $\mathcal{T}$ is obvious from the context, simply "$\mathcal{A} \wedge \mathcal{B}$".

It is well-known that in every complete lower semi-lattice, arbitrary least upper bounds exist, too. Assuming a set $\mathbb{A}$ of subalgebras of $\mathcal{T}$, the least upper bound (or *join*) $\bigvee_{\preccurlyeq,\mathcal{T}} \mathbb{A}$ is simply the greatest lower bound of the set containing those subalgebras of $\mathcal{T}$ that contain all elements of $\mathbb{A}$. (We correspondingly use "$\mathcal{A} \vee_{\mathcal{T}_{\preccurlyeq}} \mathcal{B}$" resp. "$\mathcal{A} \vee \mathcal{B}$" for binary joins.)

In addition to the existence of joins, the above theorem then also guarantees that all general lattice laws hold.

It is obvious that the above definition of joins is not so useful in the computational context — a direct definition corresponding to that of meets would be more desirable. However, the analogous definition via component-wise joins is in general not correct, as we show with a simple example: Let us define the following $\mathsf{sigB1}$-algebra $\mathcal{T}_3$:



- $\mathcal{N}^{\mathcal{T}_3} = \{0, 1, 2\}$
- $f^{\mathcal{T}_3} = \left\{ \begin{array}{lll} (0,0)\mapsto 0, & (1,0)\mapsto 2, & (2,0)\mapsto 1 \\ (0,1)\mapsto 2, & (1,1)\mapsto 1, & (2,1)\mapsto 0 \\ (0,2)\mapsto 1, & (1,2)\mapsto 0, & (2,2)\mapsto 2 \end{array} \right\}$

Now consider the two subalgebras $\mathcal{A}$ and $\mathcal{B}$ defined as follows:

- $\mathcal{N}^{\mathcal{A}} = \{0\}$ and $\mathcal{N}^{\mathcal{B}} = \{1\}$;
- $f^{\mathcal{A}} = \{(0,0) \mapsto 0\}$ and $f^{\mathcal{B}} = \{(1,1) \mapsto 1\}$.

Obviously, the carrier of their join will include both 0 and 1, but since $f^{\mathcal{T}_3}(0,1) = 2$, the carrier also needs to contain 2, and it thus turns out that $\mathcal{A} \vee \mathcal{B} = \mathcal{T}_3$.

A different effect may be observed for empty joins: consider the $\mathsf{sigC1}$-algebra $\mathcal{T}_1$ defined as follows:

- $\mathcal{N}^{\mathcal{T}_1} = \{0\}$,
- $c^{\mathcal{T}_1} = 0$.

Obviously, $\mathcal{T}_1$ is its only subalgebra, so $\bigvee_{\preccurlyeq,\mathcal{T}_1} \emptyset = \mathcal{T}_1$, i.e., the empty join has a non-empty carrier.

The construction of joins is, in fact, a special case of the well-known *closure* of carrier set families under the operations of the algebra. This closure of a family of carriers $\mathcal{X} = (s^{\mathcal{X}})_{s:\mathcal{S}}$, in the universal algebra literature often written "$\mathrm{Sg}(\mathcal{X})$" for "subuniverse generated by $\mathcal{X}$", may be defined *descriptively* as the least family of carriers which defines a subalgebra of $\mathcal{T}$ and at the same time contains $\mathcal{X}$. (For two $\Sigma$-algebras or $\Sigma$-families of carriers $\mathcal{X}$ and $\mathcal{Y}$ we write $\mathcal{X} \subseteq \mathcal{Y}$ to mean that $s^{\mathcal{X}} \subseteq s^{\mathcal{Y}}$ for all sorts $s : \mathcal{S}$.)

However, this closure may also be approximated systematically, and in finite cases generated:

**Theorem 2.3.6** [↦43, 88, 93] Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a $\Sigma$-algebra $\mathcal{T}$ be given. Let the *subalgebra closure wrt.* $\mathcal{T}$, denoted $\mathsf{SAC}_{\mathcal{T}}$ be the closure operator defined as follows: for every family of carriers $\mathcal{X} = (s^{\mathcal{X}})_{s:\mathcal{S}}$ with $\mathcal{X} \subseteq \mathcal{T}$ define the $\Sigma$-algebra $\mathsf{SAC}_{\mathcal{T}}(\mathcal{X})$ as follows:

$$\mathsf{SAC}_{\mathcal{T}}(\mathcal{X}) := \bigwedge\nolimits_{\preccurlyeq,\mathcal{T}} \{\mathcal{A} : \mathcal{T}_{\preccurlyeq} \mid \mathcal{X} \subseteq \mathcal{A}\} \ .$$

Then the carrier family of this closure may also be obtained via joins in the following way:

$$(\bigcup\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}}$$

where $\tau$ maps a $\Sigma$-family of carriers $\mathcal{X} = (s^{\mathcal{X}})_{s:\mathcal{S}}$ to another $\Sigma$-family of carriers $\tau(\mathcal{X})$, defined (slightly informally) as follows (where $\#s$ is the length of sequence $s$):

$$\tau(\mathcal{X}) = (t^{\mathcal{X}} \cup \{s : \mathcal{S}^*;\ f : s_1 \times \cdots \times s_{\#s} \to t;\ x_1 : s_1^{\mathcal{X}}; \ldots;\ x_{\#s} : s_{\#s}^{\mathcal{X}} \bullet f^{\mathcal{T}}(x_1, \ldots, x_{\#s})\})_{t:\mathcal{S}}$$

**Proof**: Let $\mathcal{Y}$ be the family of carriers of $\mathsf{SAC}_{\mathcal{T}}(\mathcal{X})$. Since $\mathcal{X} \subseteq \mathcal{Y}$ and $\mathcal{Y}$ is the carrier family of a $\Sigma$-algebra (as subalgebra of $\mathcal{T}$), it is easy to see that for all $n : \mathbb{N}$ the inclusion $\tau^n(\mathcal{X}) \subseteq \mathcal{Y}$ implies $\tau^{n+1}(\mathcal{X}) \subseteq \mathcal{Y}$, so by induction we have

$$(\bigcup\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}} \subseteq \mathcal{Y} \ .$$

Obviously, we also have $\mathcal{X} \subseteq \tau^n(\mathcal{X})$ for all $n : \mathbb{N}$, so

$$\mathcal{X} \subseteq (\bigcup\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}}$$

By definition, $(\bigcup\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}}$ also defines a subalgebra of $\mathcal{T}$, so the stated equality holds because $\mathcal{Y}$ defines the least subalgebra containing $\mathcal{X}$. □

This immediately provides us with the means to calculate joins and least elements in a subalgebra lattice:

**Theorem 2.3.7** Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a $\Sigma$-algebra $\mathcal{T}$ be given. If $\mathbb{A}$ is a subset of $\mathcal{T}_{\preccurlyeq}$, then

$$\bigvee\nolimits_{\preccurlyeq,\mathcal{T}} \mathbb{A} = \mathsf{SAC}_{\mathcal{T}}((\bigcup\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\})_{s:\mathcal{S}}) \ .$$

For two subalgebras $\mathcal{A}$ and $\mathcal{B}$ of $\mathcal{T}$, their binary join $\mathcal{A} \vee \mathcal{B}$ therefore is $\mathsf{SAC}_{\mathcal{T}}((s^{\mathcal{A}} \cup s^{\mathcal{B}})_{s:\mathcal{S}})$.

The least element in $\mathcal{T}_{\preccurlyeq}$ is $\bigvee_{\preccurlyeq,\mathcal{T}} \emptyset = \mathsf{SAC}_{\mathcal{T}}((\emptyset)_{s:\mathcal{S}})$ . □

It is also well-known that for every algebraic lattice $\mathcal{L}$ there is a (one-sorted) algebra the subalgebra lattice of which is isomorphic to $\mathcal{L}$. This implies that we cannot expect more properties to hold in general in subalgebra lattices, in particular not distributivity, for which the algebra $\mathcal{T}_3$ of page 38 already is a counterexample: Its subalgebra lattice is isomorphic to the lattice $M_5$ which is a sublattice precisely of all non-distributive modular lattices. We indicate subalgebras by their carriers:

$$\begin{tikzcd}[column sep=small]
 & \{0,1,2\} \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
\{0\} \arrow[dr, dash] & \{1\} \arrow[d, dash] & \{2\} \arrow[dl, dash] \\
 & \{\} &
\end{tikzcd}$$

The situation changes, however, when we turn to restricted classes of algebras. For our purposes, the most interesting restriction is that to unary algebras — note that all the strange effects exhibited in the examples of this section were due either to the presence of constants, or to the presence of (at least) binary operators.



## 2.4 Subalgebras in Graph Structures

We have seen that graphs can be considered as $\mathsf{sigGraph}$-algebras. One of the special properties of the signature $\mathsf{sigGraph}$ is that it is *unary*, that is, it contains only unary operators. $\Sigma$-algebras over unary signatures are also called *unary algebras*.

Since unary algebras share so many properties with graphs, they are also given a special name to reflect this:

**Definition 2.4.1** A *graph structure* is a $\Sigma$-algebra over a unary signature. □

However, we shall not strictly stick to using this name, but also use the name "unary algebra", depending mostly on the context.

In unary algebras, joins do not need application of the subalgebra closure:

**Theorem 2.4.2** [↪43] If $\mathcal{T}$ is a $\Sigma$-algebra over a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and $\mathbb{A}$ is a set of subalgebras of $\mathcal{T}$, then the join over $\mathbb{A}$ has the carrier family

$$(\bigcup\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\})_{s:\mathcal{S}} \ .$$

**Proof**: We have to show closedness of the carriers under the operations. Consider a function symbol $f : s \to t$. For every $x : \bigcup\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\}$, there is at least one $\mathcal{A} : \mathbb{A}$ with $x \in s^{\mathcal{A}}$. Therefore, $f^{\mathcal{T}}(x) = f^{\mathcal{A}}(x) \in t^{\mathcal{A}}$, and thus also $f^{\mathcal{T}}(x) \in \bigcup\{\mathcal{A} : \mathbb{A} \bullet t^{\mathcal{A}}\}$. This shows that the union of the carriers is already closed under $f^{\mathcal{T}}$, and thus under all unary operations. □

Since in unary algebras both joins and meets are defined component-wise via set joins and meets, the properties of set joins and meets propagate to subalgebras and we have:

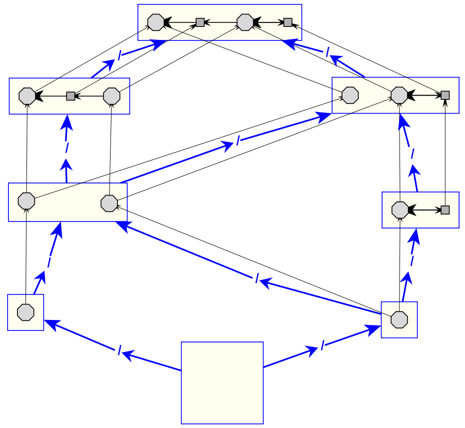
**Theorem 2.4.3** If $\mathcal{T}$ is a $\Sigma$-algebra over a unary signature $\Sigma$, then the subalgebra ordering on $\mathcal{T}_{\preccurlyeq}$ is a completely distributive lattice. □

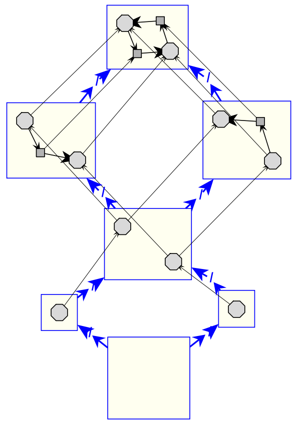
In fact, just as all algebraic lattices can be obtained as subalgebra lattices, there is a well-defined subclass of algebraic lattices that can be obtained as subalgebra lattices of unary algebras, as has been shown by Johnson and Seifert [JS67] (see also [Jón72, Theorem 3.8.9]):

**Theorem 2.4.4** For a non-trivial algebraic lattice $\mathcal{L}$ there exists a unary algebra $\mathcal{A}$ such that $\mathcal{L}$ is isomorphic to the subalgebra lattice of $\mathcal{A}$ if and only if the following conditions are satisfied:

- $\mathcal{L}$ is distributive.
- Every element of $\mathcal{L}$ is the join of join-irreducible elements.
- Each join-irreducible element of $\mathcal{L}$ contains only countably many join-irreducible elements. □

For examples of subgraph lattices, let us consider a graph with two nodes, one edge connecting them, and one loop, and a two-edge cycle:





Both of these graphs have four items, so the lattice of subsets of these graph items has sixteen elements. Their subgraph lattices, however, only have eight resp. seven elements. The largest subgraph lattice for a graph with two nodes and two edges has ten elements; that graph has two loops attached to the same node.

There are 31 different directed graphs with three nodes and three edges[1]. However, the direction of the edges is irrelevant for the structure of the subgraph lattice. Therefore these reduce to 14 different undirected graphs with three nodes and three edges. Among these, the three-edge cycle has the smallest subgraph lattice with only 18 elements; this lattice is in fact isomorphic to the free distributive lattice over three generators.

The largest subgraph lattice among the graphs with three nodes and three edges has 36 elements and results from the graph with two isolated vertices and three loops attached to one vertex. Moving one of these loops to another vertex brings the lattice size down to 30; unwinding it instead brings it down to 28. The remaining ten graphs have subgraph lattice sizes between 19 and 27 (without 20 and 23).

For the 197 different directed graphs with four vertices and four edges, at the top we have 136 and 108, most graphs have 50 to 70 subgraphs, and those which are directed variants of an undirected four-edge cycle have just 47.

All subgraph lattices are atomic, and single vertices are the atoms.

However, not all unary subalgebra lattices are atomic. A simple counterexample is the sigLoop-algebra with the set of natural numbers as carrier and the successor function as operation: the subalgebra lattice is the linear ordering of upwards-closed subsets of $\mathbb{N}$; it is an infinite downward chain with the empty set as least element, but no atom.

The source of the problem is the cycle in the signature:

**Proposition 2.4.5** If $\Sigma$ is acyclic, then the subalgebra lattices of all $\Sigma$-algebras are atomicatomic lattice. $\square$

[1] All the numbers presented here are computer-generated results.



In some contexts it is important to find the greatest subalgebra inside a given family of carriers, if that exists. In the presence of binary operators, this is not always the case; for an example consider $\mathcal{T}_3$ of page 38 and the carrier $\{0,1\}$. In unary $\Sigma$-algebras, however, such a greatest subalgebra always exists and is the result of an operator with a definition dual to that of the subalgebra closure operator $\mathsf{SAC}_{\mathcal{T}}$; at the same time, it is also the result of a fixpoint iteration dual to that of Theorem 2.3.6.

**Theorem 2.4.6** [←93] Let a unary $\Sigma$-algebra $\mathcal{T}$ be given. Let the *subalgebra kernel wrt.* $\mathcal{T}$, denoted $\mathsf{SAK}_{\mathcal{T}}$ be the operator defined as follows: for every family of carriers $\mathcal{X} = (s^{\mathcal{X}})_{s:\mathcal{S}}$ with $\mathcal{X} \subseteq \mathcal{T}$, define the $\Sigma$-algebra $\mathsf{SAK}_{\mathcal{T}}(\mathcal{X})$

$$\mathsf{SAK}_{\mathcal{T}}(\mathcal{X}) := \bigvee_{\preccurlyeq,\mathcal{T}}\{\mathcal{A} : \mathcal{T}_{\preccurlyeq} \mid \mathcal{A} \subseteq \mathcal{X}\} \ .$$

The carrier family of this kernel may also be obtained via meets in the following way:

$$(\bigcap\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}}$$

where $\tau$ maps a $\Sigma$-family of carriers $\mathcal{X} = (s^{\mathcal{X}})_{s:\mathcal{S}}$ to another $\Sigma$-family of carriers $\tau(\mathcal{X})$ defined as follows:

$$\tau(\mathcal{X}) = (\{x : s^{\mathcal{X}} \mid (\forall t : \mathcal{S}; f : \mathcal{F} \mid f : s \to t \bullet f^{\mathcal{T}}(x) \in t^{\mathcal{X}})\})_{s:\mathcal{S}}$$

**Proof**: Let $\mathbb{A} := \{\mathcal{A} : \mathcal{T}_{\preccurlyeq} \mid \mathcal{A} \subseteq \mathcal{X}\}$. Then $\bigvee_{\preccurlyeq,\mathcal{T}} \mathbb{A} \in \mathbb{A}$ since with Theorem 2.4.2 we have for every sort $s : \mathcal{S}$:

$$\begin{aligned}(\textstyle\bigvee_{\preccurlyeq,\mathcal{T}} \mathbb{A})_s &= \textstyle\bigcup\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\} \\ &= \textstyle\bigcup\{\mathcal{A} : \mathbb{A} \mid (\forall t : \mathcal{S} \bullet t^{\mathcal{A}} \subseteq t^{\mathcal{X}}) \bullet s^{\mathcal{A}}\} \subseteq \bigcup\{\mathcal{A} : \mathbb{A} \mid s^{\mathcal{A}} \subseteq s^{\mathcal{X}} \bullet s^{\mathcal{A}}\} \subseteq s^{\mathcal{X}}\end{aligned}$$

Let $\mathcal{Y}$ be the carrier family of $\mathsf{SAK}_{\mathcal{T}}(\mathcal{X})$. Since $\mathcal{Y} \subseteq \mathcal{X}$ and $\mathcal{Y}$ is the carrier family of a $\Sigma$-algebra (as subalgebra of $\mathcal{T}$), it is easy to see that for all $n : \mathbb{N}$ the inclusion $\mathcal{Y} \subseteq \tau^n(\mathcal{X})$ implies $\mathcal{Y} \subseteq \tau^{n+1}(\mathcal{X})$, so by induction we have

$$\mathcal{Y} \subseteq (\bigcap\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}} \ .$$

Obviously, we also have $\tau^n(\mathcal{X}) \subseteq \mathcal{X}$ for all $n : \mathbb{N}$, so

$$(\bigcap\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}} \subseteq \mathcal{X} \ .$$

By definition, $(\bigcap\{n : \mathbb{N} \bullet (\tau^n(\mathcal{X}))_s\})_{s:\mathcal{S}}$ also defines a subalgebra of $\mathcal{T}$, so the stated equality holds because $\mathcal{Y}$ is the carrier family of the greatest subalgebra contained in $\mathcal{X}$. $\square$

This fixpoint iteration also can be adapted to the case of arbitrary signatures; it then produces the intersection of all maximal algebras among those contained in $\mathcal{X}$. This generalised operation is *not* a kernel operator anymore since it is not monotonic: In the $\mathcal{T}_3$ example above, it contains the mappings $\{0,1\} \mapsto \{\}$ and $\{0\} \mapsto \{0\}$.

## 2.5 Partial Algebras and Weak Partial Subalgebras

For an introduction to partial algebras see [Bur86] or [Grä79, Chapt. 2].

**Definition 2.5.1** Given a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, a *partial* $\Sigma$-*algebra* $\mathcal{A}$ consists of the following items:

- a sort-indexed family $(s^{\mathcal{A}})_{s:\mathcal{S}}$ of *carrier sets*, i.e., for every sort $s : \mathcal{S}$, a set $s^{\mathcal{A}}$, and
- for every function symbol $f : \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$ a (partial) function $f^{\mathcal{A}}$ from the Cartesian product $s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}$ of the source sort carriers to $t^{\mathcal{A}}$, the target sort carrier. □

There are different subalgebra concepts for partial algebras. The natural generalisation of the subalgebra concept as usually formulated demands that operations of the subalgebra are defined for all arguments from the subalgebra's carriers where the operations of the super-algebra are defined. Such subalgebras are sometimes just called subalgebras, sometimes "closed subalgebras", for example in [BRTV96]. For this subalgebra concept, carriers of subalgebras still have to be closed, joins need to employ this closure, and the subalgebra lattices are arbitrary algebraic lattices (so need not be distributive).

An only slightly laxer concept is that of "relative subalgebras", where operations in the subalgebra need only be defined where the image of the super-algebra operation lies in the target carrier of the subalgebra.

The natural generalisation of our definition 2.3.3 of subalgebras, however, gives rise to a different subalgebra concept, which is called "weak subalgebra" already by Grätzer [Grä79]; here, operations in subalgebras are never forced to be defined:

**Definition 2.5.2** Given two partial $\Sigma$-algebras $\mathcal{A}$ and $\mathcal{T}$, we say that $\mathcal{A}$ is a *weak partial subalgebra* of $\mathcal{T}$, written $\mathcal{A} \ll \mathcal{T}$, iff

- for every sort $s : \mathcal{S}$, inclusion holds between the carriers: $s^{\mathcal{A}} \subseteq s^{\mathcal{T}}$, and
- for every function symbol $f : \mathcal{F}$, inclusion holds between its interpretations: $f^{\mathcal{A}} \subseteq f^{\mathcal{T}}$.

The set of all subalgebras of $\mathcal{T}$ is denoted by $\mathcal{T}_{\ll}$. □

The absence of the totality condition results in component-wise definitions of meets *and joins*:

**Theorem 2.5.3** If $\mathcal{T}$ is a partial $\Sigma$-algebra and $\mathbb{A}$ is a set of weak partial subalgebras of $\mathcal{T}$, then the greatest lower bound $\bigwedge_{\ll,\mathcal{T}} \mathbb{A}$ and the least upper bound $\bigvee_{\ll,\mathcal{T}} \mathbb{A}$ of $\mathbb{A}$ in the ordering $(\mathcal{T}_{\ll}, \ll)$ exist. If $\mathbb{A}$ is empty, then $\bigwedge_{\ll,\mathcal{T}} \mathbb{A} = \mathcal{T}$. Meets for non-empty $\mathbb{A}$ and arbitrary joins are defined component-wise:

$$\begin{array}{lll} \mathcal{L} := \bigwedge_{\ll,\mathcal{T}} \mathbb{A} & s^{\mathcal{L}} = \bigcap\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\} & f^{\mathcal{L}} = \bigcap\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{A}}\} \\ \mathcal{U} := \bigvee_{\ll,\mathcal{T}} \mathbb{A} & s^{\mathcal{U}} = \bigcup\{\mathcal{A} : \mathbb{A} \bullet s^{\mathcal{A}}\} & f^{\mathcal{U}} = \bigcup\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{A}}\} \end{array}$$

**Proof**: It is sufficient to show well-definedness, since the extremal properties follow from the extremal properties of the set operations.

For empty meets, the statement is obvious; otherwise, we need to show two items for well-definedness:



- closedness under the operations: if $(x_1, \ldots, x_n) \in s_1^{\mathcal{L}} \times \cdots \times s_n^{\mathcal{L}}$, then
$$f^{\mathcal{L}}(x_1, \ldots, x_n) \in \bigcap\{\mathcal{A} : \mathbb{A} \bullet t^{\mathcal{A}}\} = t^{\mathcal{L}} ,$$
and in the same way for joins.
- univalence: since
$$\begin{array}{lll} f^{\mathcal{L}} & = & \bigcap\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{A}}\} \subseteq \bigcap\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{T}}\} \subseteq f^{\mathcal{T}} \ , \text{and} \\ f^{\mathcal{U}} & = & \bigcup\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{A}}\} \subseteq \bigcup\{\mathcal{A} : \mathbb{A} \bullet f^{\mathcal{T}}\} \subseteq f^{\mathcal{T}} \ , \end{array}$$
and since $f^{\mathcal{T}}$ is univalent, $f^{\mathcal{L}}$ and $f^{\mathcal{U}}$ are univalent, too. □

Therefore, the properties of meets and joins on sets carry over to meets and joins wrt. $\ll$ on partial algebras, and it follows that $(\mathcal{T}_{\ll}, \ll)$ is a completely distributive lattice.

This analogy to the situation with total unary algebras is not by chance: there is a translation. The idea of this translation is to turn a partial algebra into a special kind of hypergraph. Elements of the carriers are considered as nodes, and elements of the interpretations of the operations — these elements are tuples of carrier elements — are considered as hyperedges. Therefore, both sorts and function symbols are sorts of the translation, and every operation of the translation associates targets with a specific kind of tentacles of a specific kind of hyperedge. These careful distinctions are necessary in order to obtain *total* unary algebras. Another possible translation would only provide one projection for every source index, and these projections would then be partial operations, defined only on hyperedges with at least that many source tentacles.

First we define the signature translation:

**Definition 2.5.4** For every signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, we define a unary signature $\mathsf{PtoU}(\Sigma) = (\mathcal{S}', \mathcal{F}', \mathsf{src}', \mathsf{trg}')$ in the following way:

- $\mathcal{S}' = \mathcal{S} + \mathcal{F}$
- $\begin{array}{lll} \mathcal{F}' & = & \{f : \mathcal{F}; s : \mathcal{S}^*; t : \mathcal{S}; i : \{1, \ldots, n\} \mid f : s_1 \times \cdots \times s_n \to t \bullet \pi_{f,i}\} \\ & \cup & \{f : \mathcal{F}; s : \mathcal{S}^*; t : \mathcal{S} \mid f : s_1 \times \cdots \times s_n \to t \bullet \rho_f\} \end{array}$
- $\mathsf{src}'(\pi_{f,i}) = f$ and $\mathsf{trg}'(\pi_{f,i} = (\mathsf{src}(f))_i$
- $\mathsf{src}'(\rho_f) = f$ and $\mathsf{trg}'(\rho_f) = \mathsf{trg}(f)$ □

The translation of partial algebras then is completely straightforward. Perhaps even surprisingly so: Function symbols are sorts in the translated signature, so have to be interpreted as carrier sets; and the interpretations we assign to them are the original interpretations as partial functions — this relies on the fact that every function is a set (more specifically, a subset of a Cartesian product).

**Definition 2.5.5** For every partial $\Sigma$-algebra $\mathcal{A}$, we define a total unary $\mathsf{PtoU}(\Sigma)$-algebra $\mathcal{U}_{\mathcal{A}}$ in the following way:

- for every $s : \mathcal{S}$, we let $s^{\mathcal{U}_{\mathcal{A}}} := s^{\mathcal{A}}$

- for every $f : \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$, we let the carrier for the sort $f$ be the set of all tuples in the interpretation of $f$ as an operation of $\mathcal{A}$:

$$f^{\mathcal{U}_\mathcal{A}} := f^\mathcal{A} = \{x_1 : s_1^\mathcal{A}, \ldots, x_n : s_n^\mathcal{A}, y : t^\mathcal{A} \mid ((x_1, \ldots, x_n), y) \in f^\mathcal{A} \bullet ((x_1, \ldots, x_n), y)\}$$

- for every $f : \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$, we let the interpretations of $\pi_{f,1}, \ldots, \pi_{f,n}$ and $\rho_f$ be the corresponding projections:

$$\pi_{f,i}^{\mathcal{U}_\mathcal{A}}((x_1, \ldots, x_n), y) = x_i \qquad\qquad \rho_f^{\mathcal{U}_\mathcal{A}}((x_1, \ldots, x_n), y) = y$$ □

Well-definedness is obvious.

This translation function $\mathcal{U}$ is obviously injective; it is even bijective, and furthermore an order-isomorphism, as is easily verified, so we have the following important result:

**Theorem 2.5.6** For every partial $\Sigma$-algebra $\mathcal{A}$, the weak subalgebra lattice $(\mathcal{A}_\ll, \ll)$ is isomorphic to the subalgebra lattice $((\mathcal{U}_\mathcal{A})_\preccurlyeq, \preccurlyeq)$. □

Therefore, all the machinery for unary total algebras can easily be made available to partial algebras via this translation. This is particularly interesting for applications to abstract state machines (ASMs, evolving algebras) [Gur91, GKOT00].

## 2.6 Pseudo-Complements

Recall that in completely upwards-distributive lattices finite meets distribute over arbitrary joins. This implies that they may also be seen as complete Heyting algebras, i.e., in completely upwards-distributive lattices the operation "$\to$" defined in the following is total:

**Definition 2.6.1** [↦86] In every lattice $(\mathcal{L}, \leqslant)$, given two elements $B$ and $A$, the *relative pseudo-complement of* $A$ *wrt.* $B$ is denoted as $A \to B$ and defined in the following way:

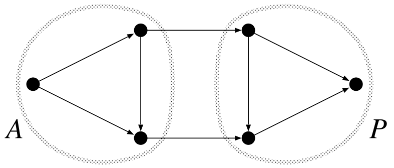
$$X \leqslant A \to B \qquad \Leftrightarrow \qquad X \wedge A \leqslant B$$

A lattice where all relative pseudo-complements exist is called *pseudo-complemented*.

If $\mathcal{L}$ has a least element, then the (absolute) *pseudo-complement* of $A$ is defined to be $A^\neg := A \to \bot$. □

In the logical view of Heyting algebras, the "relative pseudo-complement" is also called "implication".

Among subgraphs of a whole graph $\top$, the pseudo-complement of a subgraph $A$ contains all those items of $\top$, that are not adjacent to items of $A$. In the following example, we have $P = A^\neg$:




From the shape of the property it follows immediately that the relative pseudo-complement, if it exists, is uniquely determined:



**Lemma 2.6.2** [↦50] If, in an arbitrary lattice $(\mathcal{L}, \leqslant)$, for two elements $A, B : \mathcal{L}$ a relative pseudo-complement of $A$ wrt. $B$ exists, then it is uniquely determined.

**Proof**: Assume that $P$ and $P'$ are both relative pseudo-complement of $A$ wrt. $B$. Then

$$P \leqslant P \qquad \Leftrightarrow \qquad P \wedge A \leqslant B \qquad \Leftrightarrow \qquad P \leqslant P'$$

and in the same way $P' \leqslant P$, so we have $P' = P$. □

The definition of pseudo-complements defines $A \to B$ as the largest element $X : \mathcal{L}$ such that $X \wedge A \leqslant B$, if such a largest element exists; it is well-known that this may be then be obtained via a join:

**Lemma 2.6.3** If, in an arbitrary lattice $(\mathcal{L}, \leqslant)$, for two elements $A, B : \mathcal{L}$ the join

$$P := \bigvee \{X : \mathcal{L} \mid X \wedge A \leqslant B\}$$

exists and $P \wedge A \leqslant B$ holds, then $P = A \to B$.

**Proof**: Assuming $X \leqslant P$, we have, according to the assumption, $A \wedge X \leqslant A \wedge P \leqslant B$ . On the other hand, assuming $X \wedge A \leqslant B$ we obtain

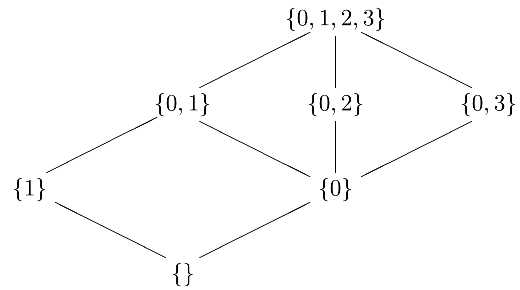
$$X \leq X \vee \bigvee \{X : \mathcal{L} \mid X \wedge A \leqslant B\} = \bigvee \{X : \mathcal{L} \mid X \wedge A \leqslant B\} = P \ ,$$

so $P$ fulfils the defining property for the relative pseudo-complement of $A$ wrt. $B$. □

As an example that even in the absence of distributivity there may be pseudo-complements, we define a sigB1-algebra $\mathcal{T}_\mathrm{d}$:

- $\mathcal{N}^{\mathcal{T}_\mathrm{d}} = \{0, 1, 2, 3\}$
- $f^{\mathcal{T}_\mathrm{d}} = \left\{ \begin{array}{llll} (0,0)\mapsto 0, & (1,0)\mapsto 1, & (2,0)\mapsto 2, & (3,0)\mapsto 3 \\ (0,1)\mapsto 1, & (1,1)\mapsto 1, & (2,1)\mapsto 0, & (3,1)\mapsto 0 \\ (0,2)\mapsto 2, & (1,2)\mapsto 3, & (2,2)\mapsto 2, & (3,2)\mapsto 0 \\ (0,3)\mapsto 3, & (1,3)\mapsto 2, & (2,3)\mapsto 1, & (3,3)\mapsto 3 \end{array} \right\}$

The subalgebra lattice of $\mathcal{T}_\mathrm{d}$ is not distributive. Nevertheless, the subalgebras $\{0\}$, $\{0,2\}$, and $\{0,3\}$ have $\{1\}$ as their pseudo-complement, and $\{0,1\}$ has the empty subalgebra $\bot$ as its pseudo-complement. $\bot$ and $\top$ are complements of each other; only $\{1\}$ has no pseudo-complement.



**Lemma 2.6.4** [←50] If, in a completely upwards-distributive lattice, for two elements $A, B : \mathcal{L}$ the join $P := \bigvee\{X : \mathcal{L} \mid X \wedge A \leqslant B\}$ exists, then $P \wedge A \leqslant B$.

**Proof**: We can show this property directly:

$$\begin{array}{lcll} P \wedge A & = & (\bigvee\{X : \mathcal{L} \mid X \wedge A \leqslant B\}) \wedge A & \\ & = & \bigvee\{X : \mathcal{L} \mid X \wedge A \leqslant B \bullet X \wedge A\} & \text{completely upwards-distributive} \\ & \leqslant & \bigvee\{X : \mathcal{L} \mid X \wedge A \leqslant B \bullet B\} \leqslant B & \end{array}$$ □

These two lemmata together ensure existence of pseudo-complements in particular in sub-algebra lattices for unary algebras:

**Theorem 2.6.5** [←50] In a completely upwards-distributive complete lattice, for every two elements $A, B : \mathcal{L}$ the relative pseudo-complement exists and obeys the following equality:

$$A \rightarrow B = \bigvee\{X : \mathcal{L} \mid X \wedge A \leqslant B\} \ .$$ □

Remember that an element $C$ is a *complement* of an element $A$ iff $C \wedge A = \bot$ and $C \vee A = \vee$; a lattice where every element has a complement is called a *Boolean lattice*. In Boolean lattices, the complement of an element $A$ is written $\overline{A}$, and we have $A \rightarrow B = \overline{A} \vee B$.

In general, the pseudo-complement coincides with the complement where the latter exists:

**Lemma 2.6.6** [←55, 58] If $A$ has a complement $C$ in a distributive bounded lattice $(\mathcal{L}, \leqslant, \top, \bot)$, then $C$ also is the pseudo-complement of $A$.

**Proof**: The proof is a slight variant of the proof of uniqueness of complements: Assuming $X \leqslant C$, the complement properties immediately give us $X \wedge A \leqslant C \wedge A = \bot$ .
On the other hand, whenever $X \wedge A \leqslant \bot$, then lattice distributivity gives us:

$$X = X \wedge \top = X \wedge (C \vee A) = (X \wedge C) \vee (X \wedge A) = (X \wedge C) \vee \bot = (X \wedge C) \ ,$$

and this implies $X \leqslant C$. Therefore, $C$ is the pseudo-complement of $A$. □

It turns out that the pseudo-complement, applied twice, give rise to a very useful closure operator, which on graphs corresponds to the concept of "induced graph". For any given subgraph $A$ of a graph $T$, the induced graph is the graph containing precisely the vertices of $A$ and precisely those edges of $T$ that are incident only with vertices of $A$.

How can we arrive at an algebraic characterisation of the induced subgraph? Obviously, it is the maximal subgraph of $T$ that does not contain any vertices outside $A$. And it not only contains no vertices outside $A$, it also cannot contain any edges incident only with those vertices. Therefore it is safe to say that the subgraph induced by $A$ is the maximal subgraph that does not contain any items of the pseudo-complement of $A$. In other words, the subgraph induced by $A$ is the pseudo-complement of the pseudo-complement of $A$:

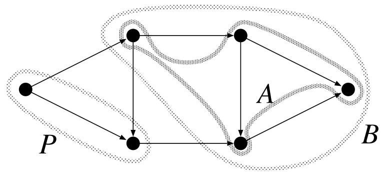


**Definition 2.6.7** In a lattice $(\mathcal{L}, \leqslant)$ with least element $\bot$, the element

$$\mathsf{Induced}(A) := (A^{\neg})^{\neg} \ ,$$

if it exists, is called the element *induced by* $A$. □

In the following drawing, we have $P = A^{\neg}$ and $B = P^{\neg} = \mathsf{Induced}(A)$:

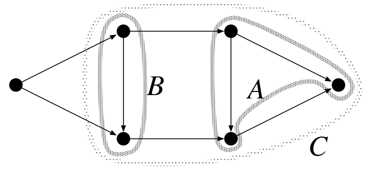



Since in a pseudo-complemented lattice we always have $A^{\neg\neg\neg} = A^{\neg}$, every pseudo-complement is induced by itself.

In a pseudo-complemented lattice, elements that are pseudo-complements of other elements are called *skeletal* elements by Grätzer [Grä78, p. 112], and the *skeleton* of a lattice is the set of all skeletal elements. We shall, however, employ the name "regular", which is frequently used in the contexts of Heyting algebras and topologies. We also let our definition be applicable to arbitrary lattices:

**Definition 2.6.8** [←63] In a lattice $(\mathcal{L}, \leqslant)$, an element $R : \mathcal{L}$ is called *regular* iff $R^{\neg\neg}$ exists and is equal to $R$. □

It is well-known that for every pseudo-complemented lattice the regular elements form a Boolean lattice[2], which is, however, not a sublattice: In that lattice of regular elements, the union of two elements $A$ and $B$ may be calculated via the pseudo-complement as $(A^{\neg} \wedge B^{\neg})^{\neg}$; this works even for pseudo-complemented semi-lattices. In lattices, one may alternatively apply the closure operator of double pseudo-complementation to obtain $\mathsf{Induced}(A \vee B)$. This can be seen nicely in graphs:




Here we have $C = \mathsf{Induced}(A \vee B)$; of the three edges that $C$ has more than $A \vee B$, only one is contained in $\mathsf{Induced}(A) \vee \mathsf{Induced}(B)$.

**Lemma 2.6.9** [←128, 180] If the relative pseudo-complement exists, then $B \leqslant A \rightarrow B$.

**Proof**: With the definition of relative pseudo-complements, $B \leqslant A \rightarrow B \Leftrightarrow A \wedge B \leqslant B$. □

[2] see e.g. [Grä78, Theorem I.6.4]

## 2.7 Semi-Complements

Since join-completeness of a lattice also implies meet-completeness, one may be tempted to ask for a dual Heyting algebra.

Although it is well-known that complete downwards-distributivity does not follow from complete upwards-distributivity, we have seen that subalgebra lattices for graph structures are not only completely upwards-distributive, but also completely downwards-distributive complete lattices.

Therefore, in the subalgebra lattices of graph structures, the dual to the relative pseudo-complement is defined, too. Nevertheless we first introduce it as a partial operation in arbitrary lattices, just like the pseudo-complement. Names proposed in the literature for this "dual pseudo-complement" include "co-implication" [Wol98] (for co-Heyting algebras), and "supplement" [But98]; it also occurs as operator "–" in [BdM97, Ex. 4.30]. Since the lattice-theoretic aspects of complementation are closer to our considerations than logical aspects, we shall not use the otherwise elegant name "co-implication". We decide to use "semi-complement", since here, "complement" may be understood on the one hand as the lattice-theoretical *terminus technicus*, so that semi-complements only share half of the properties of proper complements (while pseudo-complements share the other half), and on the other hand also in the more literal sense as "filling up together" — semi-complements do this, while pseudo-complements don't:

**Definition 2.7.1** [←87] In every lattice $(\mathcal{L}, \leqslant)$, given two elements $T, A : \mathcal{L}$, the *relative semi-complement of $A$ wrt. $T$* is denoted as $T \setminus A$ and defined in the following way:

$$T \setminus A \leqslant X \qquad \Leftrightarrow \qquad T \leqslant X \vee A \qquad\qquad \text{for all } X : \mathcal{L}.$$

A lattice where all relative semi-complements exist is called *semi-complemented*.

If $\mathcal{L}$ has a greatest element $\top$, then the (absolute) *semi-complement of $A$* is defined as $A^{\sim} := \top \setminus A$. □

We collect the dualisations of Lemmata 2.6.2 to 2.6.4 and Theorem 2.6.5 together into a single lemma:

**Lemma 2.7.2** [←54] Let an arbitrary lattice $(\mathcal{L}, \leqslant)$ and two elements $A, T : \mathcal{L}$ be given.

(i) If a relative semi-complement of $A$ wrt. $T$ exists, then it is uniquely determined, and $T \setminus A \leqslant T$.

(ii) If the meet $S := \bigwedge\{X : \mathcal{L} \mid T \leqslant X \vee A\}$ exists and $T \leqslant S \vee A$ holds, then this meet is the relative semi-complement of $A$ wrt. $T$, i.e., $S = T \setminus A$.

(iii) If the lattice $(\mathcal{L}, \leqslant)$ is completely downwards-distributive, and if the meet $S := \bigwedge\{X : \mathcal{L} \mid T \leqslant X \vee A\}$ exists, then $T \leqslant S \vee A$.

(iv) If $(\mathcal{L}, \leqslant)$ is completely downwards-distributive complete lattice, then $T \setminus A$ exists and obeys the following equality:

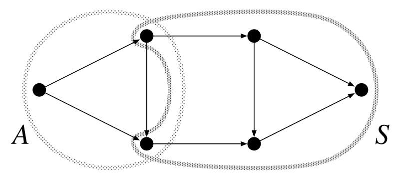
$$T \setminus A = \bigwedge\{X : \mathcal{L} \mid T \leqslant X \vee A\}$$ □



In Boolean lattices, we have $T \setminus A = T \wedge \overline{A}$, so a reading "$T$ without $A$" seems to have a certain intuitive justification — the relative semi-complement takes $A$ away from $T$ as far as possible without hurting other parts of $T$.

This is further illustrated by a concrete example taken again from subgraph lattices.

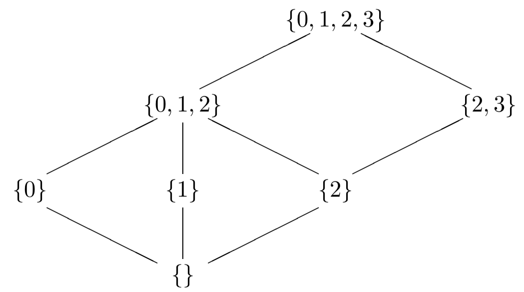
Among subgraphs of a whole graph $\top$, the pseudo-complement of a subgraph $A$ contains all those items of $\top$, that are not items of $A$ or that are adjacent to items outside $A$. In the following example, we have $S = \top \setminus A$:




For an example of a lattice where not all semi-complements exist, we extend the sigB1-algebra $\mathcal{T}_3$ from page 38 by adding a fourth element 3, yielding the sigB1-algebra $\mathcal{T}_4$:

- $\mathcal{N}^{\mathcal{T}_4} = \{0, 1, 2, 3\}$
- $f^{\mathcal{T}_4} = \left\{ \begin{array}{llll} (0,0)\mapsto 0, & (1,0)\mapsto 2, & (2,0)\mapsto 1, & (3,0)\mapsto 0 \\ (0,1)\mapsto 2, & (1,1)\mapsto 1, & (2,1)\mapsto 0, & (3,1)\mapsto 1 \\ (0,2)\mapsto 1, & (1,2)\mapsto 0, & (2,2)\mapsto 2, & (3,2)\mapsto 2 \\ (0,3)\mapsto 0, & (1,3)\mapsto 1, & (2,3)\mapsto 2, & (3,3)\mapsto 2 \end{array} \right\}$

This has the following subalgebra lattice:




From this drawing it is immediately clear that the situation here is perfectly dual to that with $\mathcal{T}_d$ on page 47: $\{0\}$, $\{1\}$, and $\{0,1,2\}$ have $\{2,3\}$ as semi-complement; $\{2\}$ has $\top$; and $\top$ and $\bot$ are complements of each other; only $\{2,3\}$ has no semi-complement.

Mostly for exhibiting the usual way of arguing with semi-complements, we prove a few properties that shall be useful later on. All these are obvious as duals of well-known properties of pseudo-complements. However, since we shall mostly need semi-complements, we hope that the reader will find these properties and their proofs instructive.

First of all, the relative semi-complement is monotonic in its first argument, and antitonic in the second:

**Lemma 2.7.3** [←52, 54, 63, 133] For all $A, B, T, U : \mathcal{L}$ in a lattice $(\mathcal{L}, \leqslant)$, subject to the existence of the semi-complements, the following hold:

(i) if $T \leqslant U$, then $T \smallsetminus A \leqslant U \smallsetminus A$;

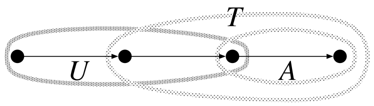
(ii) if $A \leqslant B$, then $T \smallsetminus B \leqslant T \smallsetminus A$.

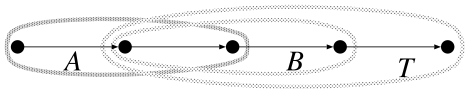
**Proof**: For all $X : \mathcal{L}$ we have:

(i) $$U \smallsetminus A \leqslant X \quad\Leftrightarrow\quad U \leqslant X \vee A \quad\Rightarrow\quad T \leqslant X \vee A \quad\Leftrightarrow\quad T \smallsetminus A \leqslant X$$

(ii) $$T \smallsetminus A \leqslant X \quad\Leftrightarrow\quad T \leqslant X \vee A \quad\Rightarrow\quad T \leqslant X \vee B \quad\Leftrightarrow\quad T \smallsetminus B \leqslant X$$ □

As illustrations why these implications are not equivalences, consider the following sub-graph counterexamples to the opposite implications (which directly correspond to simple set-theoretic examples):




On the left, we have $T \smallsetminus A \leqslant U = U \smallsetminus A$, but $T \nleqslant U$. On the right, we have $T \smallsetminus B \leqslant T \smallsetminus A$, but $A \nleqslant B$.

**Lemma 2.7.4** In a lattice where $T \smallsetminus A$ exists, if $A \leqslant T \smallsetminus A$, then $T \smallsetminus A = T$.

**Proof**: $T \smallsetminus A \leqslant T \leqslant (T \smallsetminus A) \vee A = T \smallsetminus A$. □

We already have seen the duals of $A^{\sim\sim} \leq A$ and $A^{\sim\sim\sim} = A^{\sim}$; we easily obtain the generalisations to relative semi-complements:

**Lemma 2.7.5** [←64, 65] For all $T, A, B : \mathcal{L}$ in a lattice $(\mathcal{L}, \leqslant)$, we have, subject to existence of the semi-complements:

(i) $T \smallsetminus T = \bot$ and $T \smallsetminus \bot = T$

(ii) if $A \leqslant T$, then $T \smallsetminus (T \smallsetminus A) \leqslant A$

(iii) if $A \leqslant T$, then $T \smallsetminus (T \smallsetminus (T \smallsetminus A)) = T \smallsetminus A$

**Proof**: (i) is obvious from the definition of relative semi-complements.

(ii) Assume $A \leqslant T$. For all $X : \mathcal{L}$, Lemma 2.7.3.ii) gives us:

$$A \leqslant X \quad\Rightarrow\quad T \smallsetminus X \leqslant T \smallsetminus A \quad\Leftrightarrow\quad T \leqslant X \vee (T \smallsetminus A) \quad\Leftrightarrow\quad T \smallsetminus (T \smallsetminus A) \leqslant X$$

(iii) Assume $A \leqslant T$. From $T \smallsetminus (T \smallsetminus A) \leqslant A$ follows $T \smallsetminus (T \smallsetminus (T \smallsetminus A)) \geqslant T \smallsetminus A$ via Lemma 2.7.3.ii), and (ii) directly implies $T \smallsetminus (T \smallsetminus (T \smallsetminus A)) \leqslant T \smallsetminus A$. □

There is a number of useful interactions with lattice joins and meets:



**Lemma 2.7.6** [←58, 61, 62, 64, 65, 132] In a lattice $(\mathcal{L}, \leqslant)$, for all elements $T, A, B : \mathcal{L}$ and all sets of elements $\mathcal{U} : \mathbb{P}(\mathcal{L})$, the following properties hold, subject to the existence of the semi-complements:

(i) $(U \vee T) \smallsetminus A = (U \smallsetminus A) \vee (T \smallsetminus A)$

(ii) If $(\mathcal{L}, \leqslant)$ is complete, then: $(\bigvee \mathcal{U}) \smallsetminus A = \bigvee \{U : \mathcal{U} \bullet U \smallsetminus A\}$

(iii) $T \smallsetminus (A \vee B) = (T \smallsetminus A) \smallsetminus B$

(iv) $A \vee (T \smallsetminus A) = A \vee T$

**Proof**: (i) $$\begin{aligned} (U \vee T) \smallsetminus A \leqslant X &\Leftrightarrow U \vee T \leqslant X \vee A \\ &\Leftrightarrow U \leqslant X \vee A \quad\text{and}\quad T \leqslant X \vee A \\ &\Leftrightarrow U \smallsetminus A \leqslant X \quad\text{and}\quad T \smallsetminus A \leqslant X \\ &\Leftrightarrow (U \smallsetminus A) \vee (T \smallsetminus A) \leqslant X \end{aligned}$$

(ii) $$\begin{aligned} (\textstyle\bigvee \mathcal{U}) \smallsetminus A \leqslant X &\Leftrightarrow \textstyle\bigvee \mathcal{U} \leqslant X \vee A &&\Leftrightarrow \forall U : \mathcal{U} \bullet U \leqslant X \vee A \\ &\Leftrightarrow \forall U : \mathcal{U} \bullet U \smallsetminus A \leqslant X &&\Leftrightarrow \textstyle\bigvee \{U : \mathcal{U} \bullet U \smallsetminus A\} \leqslant X \end{aligned}$$

(iii) $T \smallsetminus (A \vee B) \leqslant X \Leftrightarrow T \leqslant X \vee A \vee B \Leftrightarrow T \smallsetminus A \leqslant X \vee B \Leftrightarrow (T \smallsetminus A) \smallsetminus B \leqslant X$

(iv) $A \vee (T \smallsetminus A) \geqslant T$ by definition of the semi-complement, and $A \vee (T \smallsetminus A) \geqslant A$ trivially. The opposite inclusion follows from $T \smallsetminus A \leqslant T$. □

**Lemma 2.7.7** [←54, 58, 60, 61, 64] In a distributive lattice $(\mathcal{L}, \leqslant)$, for all elements $T, A, B : \mathcal{L}$, the following properties hold, subject to the existence of the semi-complements:

(i) $T \smallsetminus (A \wedge B) = (T \smallsetminus A) \vee (T \smallsetminus B)$

(ii) $T \smallsetminus (A \wedge T) = T \smallsetminus A$

(iii) $(T \wedge A) \vee (T \smallsetminus A) = T$

**Proof**: (i) $$\begin{aligned} T \smallsetminus (A \wedge B) \leqslant X &\Leftrightarrow T \leqslant X \vee (A \wedge B) \\ &\Leftrightarrow T \leqslant (X \vee A) \wedge (X \vee B) \\ &\Leftrightarrow T \leqslant X \vee A \quad\text{and}\quad T \leqslant X \vee B \\ &\Leftrightarrow T \smallsetminus A \leqslant X \quad\text{and}\quad T \smallsetminus B \leqslant X \\ &\Leftrightarrow (T \smallsetminus A) \vee (T \smallsetminus B) \leqslant X \end{aligned}$$

(ii) Using (i): $T \smallsetminus (A \wedge T) = (T \smallsetminus A) \vee (T \smallsetminus T) = (T \smallsetminus A) \vee \bot = T \smallsetminus A$

(iii) $(T \wedge A) \vee (T \smallsetminus A) = (T \vee (T \smallsetminus A)) \wedge (A \vee (T \smallsetminus A)) = T \wedge (A \vee T) = T$ □

**Lemma 2.7.8** [←55] In a distributive lattice $(\mathcal{L}, \leqslant)$ with greatest element $\top$, for all elements $A, B : \mathcal{L}$, the following properties hold, subject to the existence of the semi-complement:

(i) $B \smallsetminus A \leqslant A^{\sim} \wedge B$

(ii) $B \smallsetminus A = A^{\sim} \wedge B$ iff $A^{\sim} \wedge A \wedge B \leqslant B \smallsetminus A$.

**Proof**:

(i) With 2.7.2.i) we have $B \setminus A \leqslant B$, and with 2.7.3.i) also $B \setminus A \leqslant \top \setminus A = A^{\sim}$.

(ii) Because of (i), we only need to consider one inclusion:

$$\begin{array}{lll} & A^{\sim} \wedge B \leqslant B \setminus A & \\ \Leftrightarrow & A^{\sim} \wedge ((B \wedge A) \vee (B \setminus A)) \leqslant B \setminus A & \text{Lemma 2.7.7.iii)} \\ \Leftrightarrow & (A^{\sim} \wedge B \wedge A) \vee (A^{\sim} \wedge (B \setminus A)) \leqslant B \setminus A & \text{distributivity} \\ \Leftrightarrow & A^{\sim} \wedge B \wedge A \leqslant B \setminus A \quad \text{and} \quad A^{\sim} \wedge (B \setminus A) \leqslant B \setminus A & \\ \Leftrightarrow & A^{\sim} \wedge A \wedge B \leqslant B \setminus A & \end{array}$$

□

Here is a fact that may at first sight seem somewhat surprising: the meet between an element and its relative semi-complement is inseparable from the context:

**Lemma 2.7.9** [←63, 65] For all $T, A : \mathcal{L}$ in a distributive lattice $(\mathcal{L}, \leqslant)$, we have, subject to existence of the semi-complements:

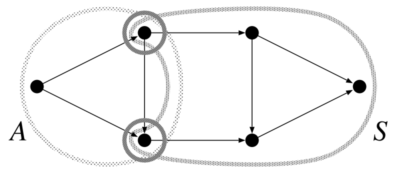
$$T = T \setminus ((T \setminus A) \wedge A)$$

**Proof**: For all $X : \mathcal{L}$ we have:

$$\begin{array}{lllll} & T \setminus ((T \setminus A) \wedge A) \leqslant X & & & \\ \Leftrightarrow & T \leqslant X \vee ((T \setminus A) \wedge A) & & & \text{semi-complement} \\ \Leftrightarrow & T \leqslant (X \vee (T \setminus A)) \wedge (X \vee A) & & & \text{distributivity} \\ \Leftrightarrow & T \leqslant X \vee (T \setminus A) & \text{and} & T \leqslant X \vee A & \text{definition of meet} \\ \Leftrightarrow & T \leqslant X \vee (T \setminus A) & \text{and} & T \setminus A \leqslant X & \text{semi-complement} \\ \Leftrightarrow & T \leqslant X & \text{and} & T \setminus A \leqslant X & \text{definition of join} \\ \Leftrightarrow & T \leqslant X & & & T \setminus A \leqslant T \end{array}$$

□

In order to make this more intuitive, consider again subgraph lattices, and for simplicity assume $T = \top$. Then the intersection $(T \setminus A) \wedge A = A^{\sim} \wedge A$, i.e., the *border* between $A$ and its semi-complement ($S := A^{\sim}$ in the following drawing), will always consist entirely of non-isolated vertices.




But non-isolated vertices cannot be taken away without taking away the incident edges, too, so the semi-complement of such a border always is the whole graph.

Finally, we show two simple (dual) properties connecting semi-complements with pseudo-complements:



**Lemma 2.7.10** [←111, 180] In a completely distributive lattice $(\mathcal{L}, \leqslant)$ with top element $\top$ and bottom element $\bot$, the following inclusions always hold:

$$\begin{array}{lcl} A \rightarrow B & \leqslant & A^{\sim} \vee B \\ A^{\neg} \wedge B & \leqslant & B \setminus A \end{array}$$

**Proof**: For every element $X : \mathcal{L}$ the following implication chain holds:

$$\begin{array}{rcl} X \leqslant A \rightarrow B & \Leftrightarrow & X \wedge A \leqslant B \\ & \Rightarrow & A^{\sim} \vee (X \wedge A) \leqslant A^{\sim} \vee B \\ & \Leftrightarrow & (A^{\sim} \vee X) \wedge (A^{\sim} \vee A) \leqslant A^{\sim} \vee B \\ & \Leftrightarrow & (A^{\sim} \vee X) \wedge \top \leqslant A^{\sim} \vee B \\ & \Leftrightarrow & A^{\sim} \vee X \leqslant A^{\sim} \vee B \\ & \Rightarrow & X \leqslant A^{\sim} \vee B \end{array}$$

For clarity, we spell out the dual argument, too:

$$\begin{array}{rcl} B \setminus A \leqslant X & \Leftrightarrow & B \leqslant X \vee A \\ & \Rightarrow & A^{\neg} \wedge B \leqslant A^{\neg} \wedge (X \vee A) \\ & \Leftrightarrow & A^{\neg} \wedge B \leqslant (A^{\neg} \wedge X) \vee (A^{\neg} \wedge A) \\ & \Leftrightarrow & A^{\neg} \wedge B \leqslant (A^{\neg} \wedge X) \vee \bot \\ & \Leftrightarrow & A^{\neg} \wedge B \leqslant A^{\neg} \wedge X \\ & \Rightarrow & A^{\neg} \wedge B \leqslant X \end{array}$$

□

Together with Lemma 2.7.8.i) and its dual, we can extend this to two-sided approximations of relative pseudo- and semi-complements:

$$\begin{array}{lclcl} A^{\neg} \vee B & \leqslant & A \rightarrow B & \leqslant & A^{\sim} \vee B \\ A^{\neg} \wedge B & \leqslant & B \setminus A & \leqslant & A^{\sim} \wedge B \end{array}$$

Setting $B$ to $\bot$ in the first line, or to $\top$ in the second line shows that pseudo-complements are contained in semi-complements:

$$A^{\neg} \leqslant A^{\sim}$$

Finally, we explicitely state the dual of Lemma 2.6.6 — the dualised proof would not provide any valuable insights, so we omit it.

**Lemma 2.7.11** If $A$ has a complement $C$ in a distributive bounded lattice $(\mathcal{L}, \leqslant, \top, \bot)$, then $C$ also is the semi-complement of $A$. □

## 2.8 Naïve Graph Rewriting

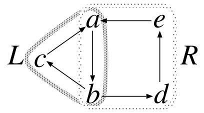
The key property of the relative semi-complement $A \setminus R$ in its application to graphs is that it does not remove interface nodes between $R$ and the part of $A$ that is the result of $A \setminus R$. In the context of graph rewriting, the semi-complement may therefore be used to delete an occurrence of the left-hand side of a rule, and these interface nodes are then the anchors for attaching a copy of the right-hand side.

With the lattice-theoretic treatment of graphs and graph structures introduced in this chapter, it is possible to specify such a rewriting step without a single mention of the words "node" and "edge".

Indeed, we would not even have to say "graph", but could instead say "element of a partial order" — we cannot operate in a single lattice here, but need the partial order of all graphs (or unary $\Sigma$-algebras). In that partial order, every set containing all elements that are smaller than a particular element is a semi-complemented lattice, and this is all we need.

However, for better understandability we still use the words "graph" and "subgraph".

We define a *rule* $(U, L, R)$ to be a graph $U$ together with two subgraphs $L$ and $R$, the *left-hand side* and the *right-hand side*, that share a common subgraph $L \wedge R$ that we shall call $G$ for *gluing graph*.




An application of such a rule, transforming an *application graph* $A$ into a result graph $B$, now proceeds as follows:

- find an isomorphic image $(U', L', R')$ of the original rule such that
  - the left-hand side is contained in the application graph: $L' \leqslant A$.
  - the right-hand side is compatible with the *host graph*

    $$H := A \setminus L' \ ,$$

    that is, the join $H \vee R'$ exists in the partial order of all graphs with the subgraph ordering, and the right-hand side overlaps with the host graph only in the gluing part:

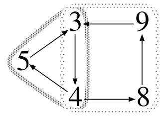
    $$H \wedge R' \ \leqslant \ L' \wedge R' \ .$$

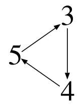
- The result of the rule application is then the join of the host graph with the right-hand side:

  $$B := H \vee R'$$

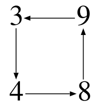
Here we show an example application of the above rule:

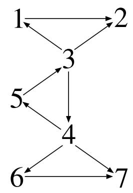


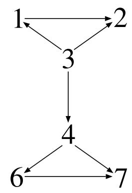
$U' =$ 

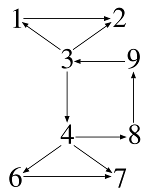

$L' =$ 


$G' =$ 


$R' =$ 


$A =$ 


$H =$ 


$B =$ 


Although this rewriting concept is very simple, it has a few significant draw-backs:

- There are no rules that perform "vertex amalgamation", that is, that would redirect all edges incident with two vertices in $A$ to a single vertex in $B$.
- Rules have to match *precisely*, that is, it is not possible for one vertex in $A$ to take on the rôles of more than one vertex in $L$.

Besides these more practical considerations, which need not be a problem in all contexts, there is also a problem with methodological hygiene:

- This rewriting concept is not purely lattice-theoretic: We needed the possibility to produce isomorphic images of the rule.

All these reasons together suggest that we need to consider graph matchings in some shape. This implies that we have to leave the simple lattice-theoretic framework. In the next chapter, we therefore move into a category-theoretic context, where morphisms serve as abstractions from graph matchings.

The conceptual framework established in the current chapter will then reappear in different guises, and will still be central to the success of establishing a relational rewriting concept.

## 2.9 Discreteness in Graph Structures

In the context of graph transformation, *discrete graphs*, i.e., graphs with no edges, play a special rôle since they act as "borders" between semi-complementary subgraphs. This is the main reason why the categoric approach to graph rewriting frequently uses discrete graphs as gluing graphs.

We now give an abstract characterisation of discreteness, recognising as characteristic of discrete objects that there, the lattice of partial identities is Boolean. Since it makes sense to talk about discrete subgraphs of arbitrary graphs, we anchor our definition of discreteness at the level of lattices (recall that the ideal generated by a lattice element $Q : \mathcal{L}$ is the sublattice containing all elements $X : \mathcal{L}$ with $X \leqslant Q$):

**Definition 2.9.1** In a lattice $(\mathcal{L}, \leqslant)$, an element $Q : \mathcal{L}$ is called *discrete* iff the ideal generated by $Q$ is a Boolean lattice. □

We show a criterion which is easier to handle in proofs:

**Lemma 2.9.2** In a distributive lattice $(\mathcal{L}, \leqslant)$ with least element $\bot$, an element $Q : \mathcal{L}$ is discrete iff for all elements $U : \mathcal{L}$ the relative semi-complement $Q \setminus U$ exists and we have

$$(Q \setminus U) \wedge U = \bot \ .$$

**Proof**: "$\Rightarrow$": If $Q$ is discrete, then for all $U : \mathcal{L}$, the meet $Q \wedge U$ is inside the ideal of $Q$ and therefore has a complement inside that ideal, say $C$. By the dual of Lemma 2.6.6 and by Lemma 2.7.7.ii) we know that

$$C = Q \setminus (Q \wedge U) = Q \setminus U \ ,$$

so, from the complement properties of $C$, we immediately obtain:

$$(Q \setminus U) \wedge U = C \wedge U = \bot \ .$$

"$\Leftarrow$": Since $(\mathcal{L}, \leqslant)$ is distributive, we only have to check that complements exist in the ideal generated by $Q$. Assume $U : \mathcal{L}$ with $U \leqslant Q$. Then, by the assumption, the relative semi-complement $S := Q \setminus U$ exists, and we have $S \wedge U = \bot$. On the other hand, the definition of semi-complements implies that $S \vee U = Q$, which is the greatest element of the ideal generated by $Q$, which shows that $S$ is a complement of $U$ in that ideal. □

It is easy to see that in subgraph lattices, the discrete elements are exactly the discrete graphs.

From the definition it is obvious that arbitrary meets of discrete elements are discrete, since for a discrete element $Q$, every element $R \leqslant Q$ is discrete, too. Even joins preserve discreteness:

**Lemma 2.9.3** [←63, 96] In a completely distributive complete lattice $(\mathcal{L}, \leqslant)$, let a set $\mathcal{Q} : \mathbb{P}(\mathcal{L})$ of elements be given. If all $Q : \mathcal{Q}$ are discrete, then $\bigvee \mathcal{Q}$ is discrete, too.

**Proof**: For $U \leqslant \bigvee \mathcal{Q}$, we have with Lemma 2.7.6.i):

$$\begin{aligned} ((\textstyle\bigvee \mathcal{Q}) \setminus U) \wedge U &= \textstyle\bigvee \{Q : \mathcal{Q} \bullet (Q \setminus U)\} \wedge U \\ &= \textstyle\bigvee \{Q : \mathcal{Q} \bullet (Q \setminus U) \wedge U\} \\ &= \textstyle\bigvee \{Q : \mathcal{Q} \bullet \bot\} \\ &= \bot \end{aligned}$$

□

This also shows that in completely distributive complete lattices, the join of *all* discrete elements is again discrete, and therefore allows us to define:



**Definition 2.9.4** For a completely distributive complete lattice $\mathcal{L}$, we denoted with $\mathbb{D}_{\mathcal{L}} : \mathcal{L}$ the *discrete base* of $\mathcal{L}$, defined as the maximal discrete element of $\mathcal{L}$:

$$\mathbb{D}_{\mathcal{L}} := \bigvee \{Q : \mathcal{L} \mid Q \text{ discrete}\}$$ □

It is already clear that a discrete element in a subgraph lattice can only contain vertices, but no edges. However, discreteness alone is not yet enough to be able to characterise those subgraphs that contain only *isolated* vertices. This needs some further preparation.

On page 54 we have seen that in graphs, the border $Q \wedge Q^{\sim}$ is always discrete. However, this is not always the case: In the four-element linear ordering $a < b < c < d$ the border for $c$ is $c$ itself:

$$c \wedge c^{\sim} = c \wedge d = c \ .$$

However, only $a$ and $b$ are discrete. Therefore we define:

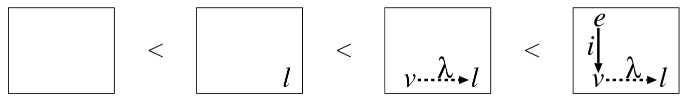
**Definition 2.9.5** The semi-complemented lattice $(\mathcal{L}, \leqslant)$ is called *border-discrete*, if for all elements $Q : (\mathcal{L}, \leqslant)$, the "border" $Q \wedge Q^{\sim}$ is discrete. □

Subgraph lattices are obviously border-discrete: borders consist only of vertices.

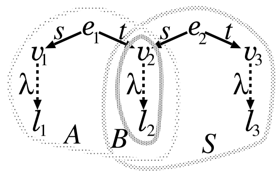
In general, subalgebra lattices for arbitrary unary algebras are border-discrete if no sort is at the same time source and target of function symbols. For counter-examples consider the following signatures:

| | |
|---|---|
| $\mathsf{sigL} :=$ **sig begin** | $\mathsf{sigGL} :=$ **sig begin** |
| **sorts:** $\mathcal{V}, \mathcal{E}, \mathcal{L}$ | **sorts:** $\mathcal{V}, \mathcal{E}, \mathcal{L}$ |
| **ops:** $i : \mathcal{E} \to \mathcal{V}$ | **ops:** $s, t : \mathcal{E} \to \mathcal{V}$ |
| $\lambda : \mathcal{V} \to \mathcal{L}$ | $\lambda : \mathcal{V} \to \mathcal{L}$ |
| **sig end** | **sig end** |

The only $\mathsf{sigL}$-algebra $D$ with $\mathcal{V}^D = \{v\}$, $\mathcal{E}^D = \{e\}$, and $\mathcal{L}^D = \{l\}$ has a subalgebra lattice which is isomorphic to the linear ordering above:




To see how this can influence borders, consider a $\mathsf{sigGL}$-algebra $T$ with carriers $\mathcal{V}^T = \{v_1, v_2, v_3\}$, $\mathcal{E}^T = \{e_1, e_2\}$, and $\mathcal{L}^T = \{l_1, l_2, l_3\}$:



For the subgraph $A$, we have the semi-complement $S := A^{\sim}$, and therefore the non-discrete border $B := A \wedge A^{\sim}$.

Subalgebra lattices for algebras over signatures like $\mathsf{sigL}$ and $\mathsf{sigGL}$ are therefore in general *not* border-discrete.

Since the presence of borders can be problematic, or at least needs special care, it is useful to characterise cases where the border is empty. An empty border indicates that the (relative) semi-complement is now a true (relative) complement:

**Definition 2.9.6** [←95] Let a semi-complemented lattice $(\mathcal{L}, \leqslant)$ and two partial elements $P, Q : \mathcal{L}$ be given.

- $P$ is called *a separable part of* $Q$ iff $P \leqslant Q$ and $(Q \setminus P) \wedge P = \bot$.
- $Q$ is called *connected* if there is, except $\bot$, no separable part of $Q$.
- The lattice $(\mathcal{L}, \leqslant)$ is called *connected* iff $\top$ is connected. □

Separable parts of $Q$ are therefore parts of $Q$ that have a true relative complement.

In graphs, there is no edge between separable parts. Therefore, our definition of connectedness gives it its conventional meaning for graphs.

Now we are ready to characterise subgraphs containing only isolated vertices as being both discrete and separable:

**Definition 2.9.7** Let a semi-complemented lattice $(\mathcal{L}, \leqslant)$ and two elements $P, Q : \mathcal{L}$ be given.

- $P$ is called *a discrete part of* $Q$ iff $P$ is discrete and $P$ is a separable part of $Q$.
- $P$ is called *a discrete part of* $(\mathcal{L}, \leqslant)$ iff $P$ is a discrete part of $\top$.
- $Q$ is called *solid* if there is, except $\bot$, no discrete part of $Q$.
- $(\mathcal{L}, \leqslant)$ is called *solid* if $\top$ is solid. □

Therefore, a subgraph $Q$ is solid iff it contains incident edges for each of its vertices, that is, iff it contains no (locally) isolated vertices. We can turn this into a useful criterion:

**Lemma 2.9.8** [←61] In a semi-complemented lattice $(\mathcal{L}, \leqslant)$, an element $Q : \mathcal{L}$ is solid iff for all discrete $R : \mathcal{L}$ we have $Q \setminus R = Q$.

**Proof**: "$\Leftarrow$": Assume that for all discrete $R : \mathcal{L}$ we have $Q \setminus R = Q$. Assume further that $P$ is a discrete part of $Q$, so $P$ is discrete, $P \leqslant Q$, and $(Q \setminus P) \wedge P = \bot$. Then the first assumption implies: $P = Q \wedge P = (Q \setminus P) \wedge P = \bot$, so $Q$ is solid.

"$\Rightarrow$": Since Lemma 2.7.7.ii) gives us: $Q \setminus R = Q \setminus (R \wedge Q)$, it is sufficient to consider only those $R$ that are contained in $Q$.

Now assume that $Q$ is solid and $R$ discrete, with $R \leqslant Q$, and define

$$Z := (Q \setminus R) \wedge R \ .$$

Then $Z \leqslant R$ and, with antitony of relative semi-complements, $Z \leqslant Q \setminus R \leqslant Q \setminus Z$. Since $Z$ is discrete because of $Z \leqslant R$, solidity of $Q$ implies $Q \setminus Z = Q$.



Now let $R' := R \setminus Z$. Then we have: $Q \setminus R = Q \setminus (Z \vee R') = (Q \setminus Z) \setminus R' = Q \setminus R'$ and

$$(Q \setminus R') \wedge R' = (Q \setminus R) \wedge R' = (Q \setminus R) \wedge R \wedge R' = Z \wedge R' = \bot$$

This shows that $R'$ is a separable part of $Q$, but since $Q$ is solid and $R'$ is discrete, it implies $R' = \bot$. Therefore, $Q \setminus R = Q \setminus R' = Q \setminus \bot = Q$. □

Every discrete part of a join can be expressed as a join of corresponding discrete parts:

**Lemma 2.9.9** In a completely distributive complete lattice $(\mathcal{L}, \leqslant)$, if the element $P : \mathcal{L}$ is a discrete part of $\bigvee \mathcal{Q}$, then

$$P = \bigvee\{Q : \mathcal{Q} \bullet Q \wedge P\}$$

and for every $Q : \mathcal{Q}$, the meet $P \wedge Q$ is a discrete part of $Q$.

**Proof**: Obviously, $P = P \wedge \bigvee \mathcal{Q} = \bigvee\{Q : \mathcal{Q} \bullet P \wedge Q\}$. Since with $P$, all $Q \wedge P$ are discrete, too, we only have to check separability:

$$\begin{array}{lll} & ((\bigvee \mathcal{Q}) \setminus P) \wedge P = \bot & \\ \Leftrightarrow & (\bigvee\{Q : \mathcal{Q} \bullet Q \setminus P\}) \wedge P = \bot & \text{Lemma 2.7.6.ii)} \\ \Leftrightarrow & \bigvee\{Q : \mathcal{Q} \bullet (Q \setminus P) \wedge P\} = \bot & \text{completely upwards distr.} \\ \Leftrightarrow & \forall Q : \mathcal{Q} \bullet (Q \setminus P) \wedge P = \bot & \\ \Leftrightarrow & \forall Q : \mathcal{Q} \bullet (Q \setminus (Q \wedge P)) \wedge (Q \wedge P) = \bot & \text{Lemma 2.7.7.ii)} \end{array}$$
□

Solidity is closed under arbotrary joins:

**Lemma 2.9.10** If for a set $\mathcal{Q} : \mathbb{P}(\mathcal{L})$ of elements of a completely distributive complete lattice $(\mathcal{L}, \leqslant)$, all elements $Q : \mathcal{Q}$ are solid, then $\bigvee \mathcal{Q}$ is solid, too.

**Proof**: Assume $P : \mathsf{PId}\,\mathcal{A}$ is discrete. Then Lemma 2.7.6.ii) and Lemma 2.9.8 yield:

$$(\bigvee \mathcal{Q}) \setminus P = \bigvee\{Q : \mathcal{Q} \bullet (Q \setminus P) = \bigvee\{Q : \mathcal{Q} \bullet Q\} = \bigvee \mathcal{Q}$$
□

This ensures that every element has a *solid part*, i.e., a maximal solid element below it, and we may define this as a join:

**Definition 2.9.11** For an element $Q : \mathcal{L}$ of a completely distributive complete lattice $(\mathcal{L}, \leqslant)$, we let $\mathsf{sol}\, Q : \mathcal{L}$ denote its *solid part*, defined as

$$\mathsf{sol}\, Q := \bigvee\{P : \mathcal{L} \mid P \leqslant Q \text{ and } P \text{ solid}\} \ .$$
□

However, a simple way to calculate the solid part is via taking the semi-complement with respect to the discrete base of the lattice:

**Lemma 2.9.12** For every element $Q : \mathcal{L}$ of a completely distributive complete lattice $(\mathcal{L}, \leqslant)$ we have $\mathsf{sol}\, Q = Q \setminus \mathbb{D}_{\mathcal{L}}$.

**Proof**: Obviously, $Q \setminus \mathbb{D}_{\mathcal{L}} \leqslant Q$.
Now assume any discrete element $R : \mathcal{L}$. Then $R \leqslant \mathbb{D}_{\mathcal{L}}$ by definition of the latter, so

$$(Q \setminus \mathbb{D}_{\mathcal{L}}) \setminus R = Q \setminus (\mathbb{D}_{\mathcal{L}} \vee R) = Q \setminus \mathbb{D}_{\mathcal{L}}$$

This shows that $Q \setminus \mathbb{D}_{\mathcal{L}}$ is solid.
Now assume any solid partial identity $P : \mathcal{L}$ with $P \leqslant Q$. Then we may use that $\mathbb{D}_{\mathcal{L}}$ is discrete, together with monotony of semi-complement in its first argument:

$$P = P \setminus \mathbb{D}_{\mathcal{L}} \leqslant Q \setminus \mathbb{D}_{\mathcal{L}}$$

This alone implies that $\mathsf{sol}\, Q \leqslant Q \setminus \mathbb{D}_{\mathcal{L}}$, and together with the above we have equality. □

The following is an important property of borders: it states that borders between $Q$ and $Q^{\sim}$ are always contained in the solid part of the semi-complement $Q^{\sim}$ — the discrete part of $Q^{\sim}$ comes into being only by the failure of $Q$ to cover a discrete part of the whole object, and never can contain part of the border.

**Lemma 2.9.13** [←97] If $Q : \mathcal{L}$ is an arbitrary element of a semi-complemented lattice with greatest element $\top$, then

$$Q \wedge Q^{\sim} \leqslant \mathsf{sol}\,(Q^{\sim}) \ .$$

**Proof**: First we have:

$$\begin{array}{rcll}
\mathsf{sol}\,(Q^{\sim}) & = & Q^{\sim} \setminus \mathbb{D}_{\mathcal{L}} & \\
 & = & (\top \setminus Q) \setminus \mathbb{D}_{\mathcal{L}} & \\
 & = & \top \setminus (Q \vee \mathbb{D}_{\mathcal{L}}) & \text{Lemma 2.7.6.iii)} \\
 & = & \top \setminus (Q \vee (\mathbb{D}_{\mathcal{L}} \setminus Q)) & \text{Lemma 2.7.6.iv)} \\
 & = & (\top \setminus Q) \setminus (\mathbb{D}_{\mathcal{L}} \setminus Q) & \text{Lemma 2.7.6.iii)} \\
 & = & Q^{\sim} \setminus (\mathbb{D}_{\mathcal{L}} \setminus Q) &
\end{array}$$

Then:

$$\begin{array}{rcll}
\mathsf{sol}\,(Q^{\sim}) \leqslant Y & \Leftrightarrow & Q^{\sim} \setminus (\mathbb{D}_{\mathcal{L}} \setminus Q) \leqslant Y & \\
 & \Leftrightarrow & Q^{\sim} \leqslant Y \vee (\mathbb{D}_{\mathcal{L}} \setminus Q) & \\
 & \Rightarrow & Q \wedge Q^{\sim} \leqslant Q \wedge (Y \vee (\mathbb{D}_{\mathcal{L}} \setminus Q)) & \\
 & \Leftrightarrow & Q \wedge Q^{\sim} \leqslant (Q \wedge Y) \vee (Q \wedge (\mathbb{D}_{\mathcal{L}} \setminus Q)) & \\
 & \Leftrightarrow & Q \wedge Q^{\sim} \leqslant (Q \wedge Y) \vee \bot & \mathbb{D}_{\mathcal{L}} \text{ discrete} \\
 & \Leftrightarrow & Q \wedge Q^{\sim} \leqslant Y &
\end{array}$$

This shows $Q \wedge Q^{\sim} \leqslant \mathsf{sol}\,(Q^{\sim})$. □

Several discrete parts form a join that continues to be a discrete part:

**Lemma 2.9.14** If for some set $\mathcal{P} : \mathbb{P}(\mathcal{L})$ of elements of a completely distributive complete lattice $(\mathcal{L}, \leqslant)$, each element $P : \mathcal{P}$ is a discrete part of $Q : \mathcal{L}$, then $\bigvee \mathcal{P}$ is a discrete part of $Q$, too.



**Proof**: Because of Lemma 2.9.3, we know that $\bigvee \mathcal{P}$ is discrete.
For every $P : \mathcal{P}$, Lemma 2.7.3.ii) gives us:

$$(Q \setminus (\bigvee \mathcal{P})) \wedge P \leqslant (Q \setminus P) \wedge P = \bot \ .$$

From this, we obtain with complete upwards distributivity:

$$(Q \setminus \bigvee \mathcal{P}) \wedge \bigvee \mathcal{P} = \bigvee \{P : \mathcal{P} \bullet (Q \setminus \bigvee \mathcal{P}) \wedge P\} = \bigvee \{P : \mathcal{P} \bullet \bot\} = \bot \ .$$ □

Taking the semi-complement with respect to a discrete element does not change a solid part:

**Lemma 2.9.15** [←97] Let a discrete element $Q : \mathcal{L}$ and an arbitrary element $R : \mathcal{L}$ be given. If there is a solid element $P : \mathcal{L}$ such that $P \leqslant R$ and $Q \wedge R \leqslant P$, then $P \setminus Q = P$ and $R \setminus Q = R$, implying in particular $Q^{\sim} = \top$.

**Proof**: Since $P$ is solid and $Q$ is discrete, we have $P \setminus Q = P$. Therefore,

$$\begin{array}{rcll}
R \setminus Q & = & (P \vee (R \setminus P)) \setminus Q & \\
 & = & (P \setminus Q) \vee ((R \setminus P) \setminus Q) & \\
 & = & P \vee (R \setminus (P \vee Q)) & \\
 & = & P \vee (R \setminus P) & Q \wedge R \leqslant P \\
 & = & R &
\end{array}$$ □

## 2.10 Coregular Parts and Base Elements

On page 60, we have seen an example of a non-discrete border; it is easy to see that that border is solid. However, Lemma 2.7.9 tells us that borders *always* are inseparable from the context, so solid substructures still need not be "significant". A "significant" part $P$ of $Q$ should not contain any part that is an "insignificant" part of $Q \setminus P$, which is the dual of regularity (Def. 2.6.8):

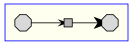
**Definition 2.10.1** Let a lattice $(\mathcal{L}, \leqslant)$ and two elements $P, Q : \mathcal{L}$ be given.

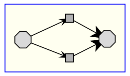
- $P$ is called *a coregular part of* $Q$ iff $P \leqslant Q$ and $Q \setminus (Q \setminus P) = P$.
- $P$ is called *coregular* iff $P$ is a coregular part of $\top$. □

In connected graphs, coregular is the same as solid. In graphs that contain isolated vertices, however, coregular parts can contain some of these. In general, coregular parts need therefore not be solid, but may have discrete parts, but only if they are discrete parts of the whole. So careful distinction of these concepts is necessary.

Coregular is, in general, not equal to regular. To see this, consider the graph consisting of a single edge incident to two vertices. The single vertices are both regular, but not coregular: one single vertex is the pseudo-complement of the other, but the semi-complement of a single vertex is the whole graph.

On the other hand, in a graph consisting of two parallel edges, the single edges are coregular, but not regular: one single edge (together with the two vertices) is the semi-complement of the other, but the pseudo-complement of a single edge is the empty subgraph.





Coregular parts are not affected by taking the semi-complement with respect to their border:

**Lemma 2.10.2** [←64] If $P$ is a coregular part of $Q$, then $P \smallsetminus (Q \smallsetminus P) = P$.

**Proof**: By direct calculation:

$$\begin{array}{rcll} P & = & Q \smallsetminus (Q \smallsetminus P) & P \text{ coregular part of } Q \\ & = & ((Q \smallsetminus P) \vee P) \smallsetminus (Q \smallsetminus P) & P \leqslant Q \\ & = & ((Q \smallsetminus P) \smallsetminus (Q \smallsetminus P)) \vee (P \smallsetminus (Q \smallsetminus P)) & \text{Lemma 2.7.6.i)} \\ & = & P \smallsetminus (Q \smallsetminus P) & \text{Lemma 2.7.5.i)} \end{array}$$

□

**Lemma 2.10.3** The relation "is a coregular part of" among elements of a completely distributive complete lattice $(\mathcal{L}, \leqslant)$ is an ordering.

**Proof**: Reflexivity follows from reflexivity of $\leqslant$ and $P \smallsetminus (P \smallsetminus P) = P \smallsetminus \bot = P$.

For transitivity assume that $P$ is a coregular part of $Q$, and $Q$ is a coregular part of $R$. Then $P \leqslant R$ by transitivity of $\leqslant$, and we first have:

$$\begin{array}{rcll} R \smallsetminus P & = & ((R \smallsetminus Q) \vee Q) \smallsetminus P & \\ & = & ((R \smallsetminus Q)) \smallsetminus P \vee (Q \smallsetminus P) & \text{Lemma 2.7.6.i)} \\ & = & (R \smallsetminus (Q \vee P)) \vee (Q \smallsetminus P) & \text{Lemma 2.7.6.iii)} \\ & = & (R \smallsetminus Q) \vee (Q \smallsetminus P) & P \leqslant Q \end{array}$$

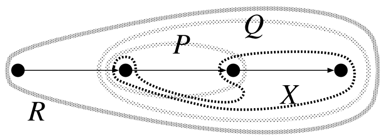
Now we can show that $P$ is a coregular part of $Q$:

$$\begin{array}{rcll} R \smallsetminus (R \smallsetminus P) & = & R \smallsetminus (R \smallsetminus (Q \wedge P)) & P \leqslant Q \\ & = & R \smallsetminus ((R \smallsetminus Q) \vee (R \smallsetminus P)) & \text{Lemma 2.7.7.i)} \\ & = & (R \smallsetminus (R \smallsetminus Q)) \smallsetminus (R \smallsetminus P) & \text{Lemma 2.7.6.iii)} \\ & = & Q \smallsetminus (R \smallsetminus P) & Q \text{ coregular part of } R \\ & = & Q \smallsetminus (Q \wedge (R \smallsetminus P)) & \text{Lemma 2.7.7.ii)} \\ & = & Q \smallsetminus (Q \wedge ((R \smallsetminus Q) \vee (Q \smallsetminus P))) & \text{see above} \\ & = & Q \smallsetminus ((Q \wedge (R \smallsetminus Q)) \vee (Q \wedge (Q \smallsetminus P))) & \text{distributivity} \\ & = & Q \smallsetminus ((Q \wedge (R \smallsetminus Q)) \vee (Q \smallsetminus P)) & Q \smallsetminus P \leqslant Q \\ & = & (Q \smallsetminus (Q \wedge (R \smallsetminus Q))) \smallsetminus (Q \smallsetminus P) & \text{Lemma 2.7.6.iii)} \\ & = & (Q \smallsetminus (R \smallsetminus Q)) \smallsetminus (Q \smallsetminus P) & \text{Lemma 2.7.7.ii)} \\ & = & Q \smallsetminus (Q \smallsetminus P) & \text{Lemma 2.10.2} \\ & = & P & P \text{ coregular part of } Q \end{array}$$

Antisymmetry directly follows from antisymmetry of $\leqslant$. □



The reason for the lengthy calculation in the proof of transitivity is the effect that, in general, $Q \wedge (R \smallsetminus P) \neq Q \smallsetminus P$, as in the following example, where $X := Q \wedge (R \smallsetminus P)$ has a discrete part, which is not contained in $Q \smallsetminus P$:




However this discrete part is a border of $Q$, and, as already explained after Lemma 2.7.9, such borders can never be globally discrete, so they disappear below the next level of relative semi-complements.

**Definition 2.10.4** Let a semi-complemented lattice $(\mathcal{L}, \leqslant)$ and two elements $P, Q : \mathcal{L}$ be given.

- $P$ is called *an essential part of* $Q$ iff $P \leqslant Q$ and $Q \smallsetminus P \neq Q$.
- $Q$ is called an *base element of* $\mathcal{L}$ iff $Q$ has no essential part except itself. □

In graphs, a base element is a subgraph either consisting of a single vertex, or of a single edge together with its adjacent vertices. Since there are no smaller subgraphs containing single edges, these already represent single edges as far as possible. Therefore, base elements are essentially single vertices and single edges, which justifies the choice of the name "element".

Obviously, vertex base elements are discrete, while edge base elements are solid.

**Lemma 2.10.5** $Q : \mathcal{L}$ is a base element iff $Q$ is join-irreducible.

**Proof**: "$\Rightarrow$": Assume that $Q$ is a base element, and that $Q = A \vee B$. Then we have $A \leqslant Q$ and $B \leqslant Q$. Assuming $A \neq Q$, we have $Q \smallsetminus A = Q$ since $A$ cannot be an essential part of $Q$. But with Lemma 2.7.6.i) we have

$$Q = Q \smallsetminus A = (A \vee B) \smallsetminus A = A \smallsetminus A \vee B \smallsetminus A = \bot \vee B \smallsetminus A = B \smallsetminus A \ .$$

Therefore, $Q \leqslant B$, and finally $Q = B$.

In the same way we obtain $Q = A$ from $B \neq Q$.

"$\Leftarrow$": Assume that $Q$ is join-irreducible, and that $P$ is an essential part of $Q$, that is, $P \leqslant Q$ and $Q \smallsetminus P \neq Q$.

Then, by the definition of relative semi-complements, and by $P \leqslant Q$ we have $Q = (Q \smallsetminus P) \vee P$. From join-irreducibility and from $Q \smallsetminus P \neq Q$ we obtain $P = Q$, which shows that $Q$ is a base element. □

The definition of essential parts mostly served to make base elements more accessible. The following lemma shows that base elements might equally have been defined via coregular parts:

**Lemma 2.10.6** $Q$ is a base element iff $Q$ has no non-empty coregular part except itself.

**Proof**: "$\Rightarrow$": this is obvious, since every non-empty coregular part is an essential part.

"$\Leftarrow$": assume $P \leqslant Q$ and $Q \smallsetminus P \neq Q$. Then $Q \smallsetminus P$ is a coregular part of $Q$ by Lemma 2.7.5.iii), and if $P \neq Q$, then $Q \smallsetminus P$ is non-empty. □

# Chapter 3

# Allegories of $\Sigma$-Algebras

The discussion of parts of graphs in the previous chapter took place in the more abstract setting of subgraph lattices, never mentioning nodes or edges in the definitions. This has the advantage that all considerations are immediately valid for a large class of graph-like structures, too.

In order to achieve the same for graph homomorphisms, we need an essentially category-theoretic setting. Since we want to be able to admit relational matching, we need an appropriate generalisation of relations, and find this in *allegories*, which are a relatively weak kind of relation categories. However, they are general enough to admit even relational homomorphisms between general $\Sigma$-algebras for a fixed, not necessarily unary, signature $\Sigma$. For this reason, the introduction of relational homomorphisms will be for algebras over arbitrary signatures, and we can show that these relational homomorphisms give rise to a certain kind of allegories.

With unary algebras, we may move to stronger restrictions of allegories, and we do this in the next chapter. Readers who are not interested in the complications arising in general algebras may skip to the next chapter, perhaps best only after heaving read the preliminaries section of this chapter up to Def. 3.1.8.

The current chapter starts with a section providing fundamental background on categories and allegories, including definitions related with direct products. We then define an abstract variant of $\Sigma$-algebras, and relational homomorphisms for them, proving the allegory properties. Finally we show how standard constructions like substructures, quotients and product structures are available in this abstract and relational setting, too.

## 3.1 Preliminaries: Categories and Allegories

We recall the definition of a category.

**Definition 3.1.1** A *category* $\mathbf{C}$ is a tuple $(Obj_{\mathbf{C}}, Mor_{\mathbf{C}}, \_ : \_ \to \_, \mathbb{I}, ;)$ where

- $Obj_{\mathbf{C}}$ is a collection of *objects*.
- $Mor_{\mathbf{C}}$ is a collection of *arrows* or *morphisms*.
- "$\_ : \_ \to \_$" is ternary predicate relating every morphism $f$ univalently with two objects $\mathcal{A}$ and $\mathcal{B}$, written $f : \mathcal{A} \to \mathcal{B}$, where $\mathcal{A}$ is called the *source* of $f$, and $\mathcal{B}$ the *target* of $f$.

  The collection of all morphisms $f$ with $f : \mathcal{A} \to \mathcal{B}$ is denoted as $\mathsf{Mor}_{\mathbf{C}}[\mathcal{A}, \mathcal{B}]$ and also called a *homset*.
- "$;$" is the binary *composition* operator, and composition of two morphisms $f : \mathcal{A} \to \mathcal{B}$ and $g : \mathcal{B}' \to \mathcal{C}$ is defined iff $\mathcal{B} = \mathcal{B}'$, and then $(f;g) : \mathcal{A} \to \mathcal{C}$; composition is associative.
- $\mathbb{I}$ associates with every object $\mathcal{A}$ a morphism $\mathbb{I}_{\mathcal{A}}$ which is both a right and left unit for composition. □





The composition operator "$;$" will bind with a higher priority than all other binary operators.

An object $\mathbb{O}$ in a category is called *initial* iff for every other object $\mathcal{A}$ there is exactly one arrow from $\mathbb{O}$ to $\mathcal{A}$. Dually, an object $\mathbb{1}$ is called *terminal* iff for every other object $\mathcal{A}$ there is exactly one arrow from $\mathcal{A}$ to $\mathbb{1}$.

**Definition 3.1.2** An *allegory* is a tuple $\mathbf{C} = (Obj_{\mathbf{C}}, Mor_{\mathbf{C}}, \_ : \_ \leftrightarrow \_, \mathbb{I}, ;, ^{\smile}, \sqcap)$ where:

- The tuple $(Obj_{\mathbf{C}}, Mor_{\mathbf{C}}, \_ : \_ \leftrightarrow \_, \mathbb{I}, ;)$ is a category, the so-called *underlying category* of $\mathbf{C}$. The morphisms are usually called *relations*.
- Every homset $\mathsf{Mor}_{\mathbf{C}}[\mathcal{A}, \mathcal{B}]$ carries the structure of a lower semi-lattice[1] with $\sqcap_{\mathcal{A},\mathcal{B}}$ for *meet*, and inclusion ordering $\sqsubseteq_{\mathcal{A},\mathcal{B}}$, all usually written without indices.
- "$^{\smile}$" is the total unary operation of *conversion* of morphisms, where for $R : \mathcal{A} \leftrightarrow \mathcal{B}$ we have $R^{\smile} : \mathcal{B} \leftrightarrow \mathcal{A}$, and the following properties hold:
$$(R^{\smile})^{\smile} = R \ , \qquad (Q;R)^{\smile} = R^{\smile};Q^{\smile} \ , \qquad (Q \sqcap Q')^{\smile} = Q^{\smile} \sqcap Q'^{\smile} \ .$$
- For all $Q : \mathcal{A} \leftrightarrow \mathcal{B}$ and $R, R' : \mathcal{B} \leftrightarrow \mathcal{C}$, *meet-subdistributivity* holds:
$$Q;(R \sqcap R') \sqsubseteq Q;R \sqcap Q;R' \ .$$
- For all $Q : \mathcal{A} \leftrightarrow \mathcal{B}$, $R : \mathcal{B} \leftrightarrow \mathcal{C}$, and $S : \mathcal{A} \leftrightarrow \mathcal{C}$, the *modal rule* holds:
$$Q;R \sqcap S \sqsubseteq (Q \sqcap S;R^{\smile});R \ .$$ □

The following basic properties are easily deduced from the definition of allegories:

- Conversion is an isotone and involutive contravariant functor: In addition to the properties from the definition, this comprises also $\mathbb{I}_A^{\smile} = \mathbb{I}_A$ and $Q \sqsubseteq Q' \Leftrightarrow Q^{\smile} \sqsubseteq Q'^{\smile}$.
- Composition is monotonic: If $Q \sqsubseteq Q'$ and $R \sqsubseteq R'$, then $Q;R \sqsubseteq Q';R'$.

From the modal rule listed among the allegory axioms, we may — using properties of conversion — obtain the other modal rule
$$Q;R \sqcap S \sqsubseteq Q;(R \sqcap Q^{\smile};S) \ ,$$
which is used by Olivier and Serrato for their axiomatisation of Dedekind categories [OS80, OS95] (see also next chapter) and there called "Dedekind formula" — however, Jacques Riguet had much earlier attached the name "Dedekind formula" to the following formula [Rig48]:
$$Q;R \sqcap S \sqsubseteq (Q \sqcap S;R^{\smile});(R \sqcap Q^{\smile};S) \ .$$
The Dedekind formula is in fact equivalent to the modal rules, see Proposition A.1.1.

Another possible variation in the axiomatisation of allegories stems from the fact that meet-subdistributivity of composition is equivalent to monotony of composition.

[1] A homset $\mathsf{Mor}_{\mathbf{C}}[\mathcal{A}, \mathcal{B}]$ may be a class in [FS90], meaning that there, allegories are not restricted to be locally small. The price of this generality, however, is that join, meet, etc. need to be characterised at a more elementary level, while we can introduce these as lattice operators.

Many standard properties of relations can be characterised in purely allegorical language:

**Definition 3.1.3** In an allegory **A**, for a relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ we define the following properties:

- $R$ is *univalent* iff $R^{\smile};R \sqsubseteq \mathbb{I}_{\mathcal{B}}$,
- $R$ is *total* iff $\mathbb{I}_{\mathcal{A}} \sqsubseteq R;R^{\smile}$,
- $R$ is *injective* iff $R;R^{\smile} \sqsubseteq \mathbb{I}_{\mathcal{A}}$,
- $R$ is *surjective* iff $\mathbb{I}_{\mathcal{B}} \sqsubseteq R^{\smile};R$,
- $R$ is a *mapping* iff $R$ is univalent and total,
- $R$ is *bijective* iff $R$ is injective and surjective.

Furthermore, we denote the subcategory of **A** that contains all objects of **A**, but only mappings as arrows with $\mathsf{Map}\,\mathbf{A}$, and that with all partial functions (i.e., univalent relations) with $\mathsf{Pfn}\,\mathbf{A}$. □

We call a relation *homogoneous* iff its source and target objects coincide. For homogeneous relations, there are a few additional properties of interest:

**Definition 3.1.4** In an allegory, for a relation $R : \mathcal{A} \leftrightarrow \mathcal{A}$ we define the following properties:

- $R$ is *reflexive* iff $\mathbb{I} \sqsubseteq R$,
- $R$ is *transitive* iff $R;R \sqsubseteq R$,
- $R$ is *symmetric* iff $R^{\smile} \sqsubseteq R$,
- $R$ is *antisymmetric* iff $R^{\smile} \sqcap R \sqsubseteq \mathbb{I}$,
- $R$ is an *equivalence relation* iff $R$ is reflexive, transitive, and symmetric,
- $R$ is an *ordering* iff $R$ is reflexive, transitive, and antisymmetric. □

For homsets that have least or greatest elements, we introduce corresponding notation:

**Definition 3.1.5** In an allegory, for two objects $\mathcal{A}$ and $\mathcal{B}$ we introduce the following notions:

- If the homset $\mathsf{Mor}_{\mathbf{C}}[\mathcal{A},\mathcal{B}]$ contains a greatest element, then this *universal relation* is denoted $\mathbb{T}_{\mathcal{A},\mathcal{B}}$.
- If the homset $\mathsf{Mor}_{\mathbf{C}}[\mathcal{A},\mathcal{B}]$ contains a least element, then this *empty relation* (or *zero relation*) is denoted $\perp\!\!\!\perp_{\mathcal{A},\mathcal{B}}$. □

For these extremal relations and for identity relations we frequently omit indices where these can be induced from the context.

In the presence of universal relations, totality of $R$ is equivalent to the condition $R;\mathbb{T} \sqsupseteq \mathbb{T}$ (see Lemma A.1.4); this is often easier to use in proofs.

In an allegory, every homset is a meet-semilattice, where binary meets interact in a particular way with composition. Analogously, if arbitrary meets exist, they should interact in a corresponding way with composition in order to make sense in the allegory structure. The following property has, to our knowledge, not been considered before:



**Definition 3.1.6** [←85] An allegory is called *locally co-complete* iff for every two objects $\mathcal{A}$ and $\mathcal{B}$ and every set $\mathcal{R}$ of relations from $\mathcal{A}$ to $\mathcal{B}$ the greatest lower bound (wrt. $\sqsubseteq$) $\bigsqcap \mathcal{R}$ of $\mathcal{R}$ exists. □

It is easily checked that in a locally co-complete allegory, for all relations $Q : \mathcal{C} \leftrightarrow \mathcal{A}$ and for every set $\mathcal{R}$ of relations from $\mathcal{A}$ to $\mathcal{B}$ the following properties hold:

- distributivity of converse: $(\bigsqcap \mathcal{R})^{\smile} \;=\; \bigsqcap\{R : \mathcal{R} \bullet R^{\smile}\}$
- subdistributivity of composition: $Q;\bigsqcap \mathcal{R} \;\sqsubseteq\; \bigsqcap\{R : \mathcal{R} \bullet Q;R\}$

Subdistributivity for composition from the right follows via conversion from that for composition from the left. If $Q$ is univalent and $\mathcal{R}$ non-empty, we even have distributivity (=) instead of subdistributivity ($\sqsubseteq$) for composition with $Q$ from the left (Lemma A.1.5).

The allegory of sets with concrete relations obviously is locally co-complete.

In locally co-complete allegories, all universal relations exist, since

$$\bigsqcap \emptyset = \mathbb{T} \ ,$$

and all empty relations exist, since

$$\bigsqcap(\mathsf{Mor}[\mathcal{A},\mathcal{B}]) = \perp\!\!\!\perp_{\mathcal{A},\mathcal{B}} \ .$$

Relations that are contained in the identity are referred to by many different names in the literature, such as "coreflexives" by Freyd and Scedrov [FS90], or "monotypes" by the group of Backhouse [ABH+92, DvGB97]. They are particularly important since they are the simplest mechanism available in all allegories that allows to characterise "parts" of objects, corresponding to subsets of sets in the allegory of sets and concrete relations. We stick to a less sophisticated name, which is also in wide-spread use in the literature (e.g. in [SHW97, DMN97]):

**Definition 3.1.7** A *partial identity* is a relation contained in an identity. For every object $\mathcal{A}$ of an allegory, we denote the set of partial identities on $\mathcal{A}$ with $\mathsf{PId}\,\mathcal{A}$. □

Partial identities arise in particular as abstractions of the concrete concepts of "domain of definition" and "range of values" of a relation (which should never be confused with the categorical concepts of source and target of a morphism):

**Definition 3.1.8** [←66] For every relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ in an allegory, we define $\mathsf{dom}\,R : \mathcal{A} \leftrightarrow \mathcal{A}$ and $\mathsf{ran}\,R : \mathcal{B} \leftrightarrow \mathcal{B}$ as:

$$\mathsf{dom}\,R := \mathbb{I}_{\mathcal{A}} \sqcap R;R^{\smile} \qquad\qquad \mathsf{ran}\,R := \mathbb{I}_{\mathcal{B}} \sqcap R^{\smile};R$$ □

For useful properties concerning partial identities, see Sect. A.2.

According to [FS90, 2.15]:

**Definition 3.1.9** [←72] An object $\mathcal{U}$ in an allegory is a *partial unit* if $\mathbb{I}_{\mathcal{U}} = \mathbb{T}_{\mathcal{U},\mathcal{U}}$. The object $\mathcal{U}$ is a *unit* if, further, every object is the source of a total morphism targeted at $\mathcal{U}$. An allegory is said to be *unitary* if it has a unit. □

We usually use the symbol "$\mathbb{1}$" for a unit object. From [FS90, 2.152] we cite the following useful properties:

- A total morphism from any object $\mathcal{A}$ to a unit $\mathbb{1}$ is always the universal relation $\mathbb{T}_{\mathcal{A},\mathbb{1}}$, and is in addition univalent, and therefore a mapping.
- In the presence of a unit, all universal relations exist, and $\mathbb{T}_{\mathcal{A},\mathcal{B}} = \mathbb{T}_{\mathcal{A},\mathbb{1}} ; \mathbb{T}_{\mathbb{1},\mathcal{B}}$.
- In the presence of a unit, there is an isomorphism between $\mathsf{PId}\,\mathcal{A}$ and $\mathsf{Mor}[\mathcal{A},\mathbb{1}]$, so that morphisms to (or from) the unit may be used as an alternative to partial identities in the rôle of identifiers of "parts" of objects.

## Direct Products

It is well-known that the self-duality of categories of relations implies that categorical sums are at the same time categorical products — in relation algebras with sets and concrete relations, categorical sums are disjoint unions.

However, Cartesian products can be axiomatised on the relational level; most approaches are rooted in homogeneous relation algebras, such as [TG87, Mad95, ABH+92, HFBV97].

We follow the "Munich approach" of Schmidt and coworkers [Sch77, Car82, BZ86, Zie88, Zie91, SS93, BHSV94], but since we need to cover the case of empty products, we use a variant that does not demand surjectivity of the projections. This slight generalisation of the original Munich approach definition shown on page 26 brings us closer to the notions of Freyd and Scedrov [FS90]; see also the discussion around Def. 5.1.2.

According to the following definition, two relations $\pi$ and $\rho$ are projections of a direct product iff, in the language of Freyd and Scedrov, they "tabulate" a universal relation.

**Definition 3.1.10** [←99] In an allegory, a *direct product* for two objects $\mathcal{A}$ and $\mathcal{B}$ is a triple $(\mathcal{P}, \pi, \rho)$ consisting of an object $\mathcal{P}$ and two *projections*, i.e., relations $\pi : \mathcal{P} \leftrightarrow \mathcal{A}$ and $\rho : \mathcal{P} \leftrightarrow \mathcal{B}$ for which the following conditions hold:

$$\pi^{\smile};\pi = \mathsf{dom}\,(\mathbb{T}_{\mathcal{A},\mathcal{B}}) \;, \qquad \rho^{\smile};\rho = \mathsf{ran}\,(\mathbb{T}_{\mathcal{A},\mathcal{B}}) \;, \qquad \pi^{\smile};\rho = \mathbb{T}_{\mathcal{A},\mathcal{B}} \;, \qquad \pi;\pi^{\smile} \sqcap \rho;\rho^{\smile} = \mathbb{I}_{\mathcal{P}} \;. \;\square$$

This definition is a monomorphic characterisation of direct product. The fact that we do not insist on surjective projections allows *empty* products.

For all direct products in allegories, the following inclusion holds:

$$P;R \sqcap Q;S \sqsupseteq (P;\pi^{\smile} \sqcap Q;\rho^{\smile});(\pi;R \sqcap \rho;S).$$

The opposite inclusion

$$P;R \sqcap Q;S \sqsubseteq (P;\pi^{\smile} \sqcap Q;\rho^{\smile});(\pi;R \sqcap \rho;S) \;,$$

named *sharpness condition* by Gunther Schmidt (see [Car82]) does not always hold, not even with surjective projections, and not even in relation algebras. A product for which this condition holds is called a *sharp product*.[2]

[2] For a (homogenous) relation algebra with an unsharp product with surjective projections, together with its history, see [Mad95]; a translation into a heterogeneous setting may be found in [KS00, Sect. 3.2].



For products of more than two objects, it is no problem to iterate the binary product construction. However, it is more elegant to have a direct characterisation for $n$-ary product. We modify the definition proposed by Desharnais [Des99] by providing for possibly non-surjective projections, again (in [Des99], the first condition reads $\pi_k^{\smile};\pi_k = \mathbb{I}_{\mathcal{A}_k}$):

**Definition 3.1.11** For a positive natural number $n : \mathbb{N}^+$ and $n$ objects $\mathcal{A}_1, \ldots, \mathcal{A}_n$, an *$n$-ary direct product* for $\mathcal{A}_1, \ldots, \mathcal{A}_n$ is a pair $(\mathcal{P}, \Pi)$ consisting of an object $\mathcal{P}$ and a family $\Pi = (\pi_i)_{i:\{1..n\}}$ of *projections* $\pi_i : \mathcal{P} \leftrightarrow \mathcal{A}_i$ fulfilling the following conditions:

- $\pi_k^{\smile};\pi_k = \mathsf{dom}\,\mathbb{T}_{\mathcal{A}_k,\mathcal{P}}$ for all $k : \{1 \mathrel{..} n\}$,
- $\pi_k^{\smile};\bigsqcap\{i : \{1 \mathrel{..} n\} \mid i \neq k \bullet \pi_i;\pi_i^{\smile}\} = \mathbb{T}_{\mathcal{A}_k,\mathcal{P}}$, for all $k : \{1 \mathrel{..} n\}$,
- $\bigsqcap\{k : \{1 \mathrel{..} n\} \bullet \pi_k;\pi_k^{\smile}\} = \mathbb{I}_{\mathcal{P}}$. □

Although Desharnais set this definition in the context of relation algebras, the paper [Des99] in fact only uses the framework of allegories with universal relations. Desharnais shows that $n$-ary direct products according to his definition are monomorphic. Unfortunately, that proof relies on the surjectivity of at least one of the projections in a non-trivial way. Since the above generalisation of the definition of Desharnais admits products with only non-surjective projections (these exist; we give an example on page 83), it is not clear whether monomorphism still holds.

Therefore, we resort to the usual mechanism of nesting products, and assuming associativity for making the precise nesting structure irrelevant. For maximal flexibility, we still provide for cases where not all products exist — this allows useful study of finite models without sacrificing honesty.

**Definition 3.1.12** In an allegory $\mathbf{D}$, a *partial choice of direct products* is a tuple $(\times, \pi, \rho)$ where

- $\times$, $\pi$ and $\rho$ are partial binary operations expecting two objects as arguments, and all three having the same domain.
- $\times$ maps, where defined, two objects $\mathcal{A}$ and $\mathcal{B}$ to an object $\mathcal{A} \times \mathcal{B}$.
- $\pi$ maps, where defined, two objects $\mathcal{A}$ and $\mathcal{B}$ to a mapping $\pi_{\mathcal{A},\mathcal{B}} : \mathcal{A} \times \mathcal{B} \to \mathcal{A}$.
- $\rho$ maps, where defined, two objects $\mathcal{A}$ and $\mathcal{B}$ to a mapping $\rho_{\mathcal{A},\mathcal{B}} : \mathcal{A} \times \mathcal{B} \to \mathcal{B}$.
- Where the components of the triple $(\mathcal{A} \times \mathcal{B}, \pi_{\mathcal{A},\mathcal{B}}, \rho_{\mathcal{A},\mathcal{B}})$ are defined, the triple is a direct product for $\mathcal{A}$ and $\mathcal{B}$.

A partial choice $(\times, \pi, \rho)$ of direct products is called *strictly associative* iff for every three objects $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, we have the equality and equivalence of definedness

$$\mathcal{A} \times (\mathcal{B} \times \mathcal{C}) = (\mathcal{A} \times \mathcal{B}) \times \mathcal{C} \;,$$

and (where the corresponding products are defined):

$$\pi_{\mathcal{A},\mathcal{B}\times\mathcal{C}} = \pi_{\mathcal{A}\times\mathcal{B},\mathcal{C}};\pi_{\mathcal{A},\mathcal{B}} \;, \qquad \rho_{\mathcal{A},\mathcal{B}\times\mathcal{C}};\pi_{\mathcal{B},\mathcal{C}} = \pi_{\mathcal{A}\times\mathcal{B},\mathcal{C}};\rho_{\mathcal{A},\mathcal{B}} \;, \qquad \rho_{\mathcal{A},\mathcal{B}\times\mathcal{C}};\rho_{\mathcal{B},\mathcal{C}} = \rho_{\mathcal{A}\times\mathcal{B},\mathcal{C}} \;.$$

An object $\mathbb{1}$ is called a *strict unit*for a partial choice $(\times, \pi, \rho)$ of direct products iff for every object $\mathcal{A}$ we have

- $\mathcal{A} \times \mathbb{1}$ and $\mathbb{1} \times \mathcal{A}$ are defined,
- $\mathcal{A} \times \mathbb{1} = \mathcal{A} = \mathbb{1} \times \mathcal{A}$,
- $\pi_{\mathcal{A},\mathbb{1}} = \mathbb{I}_{\mathcal{A}} = \rho_{\mathbb{1},\mathcal{A}}$.

An allegory **D** together with a partial strictly associative partial choice $(\times, \pi, \rho)$ of sharp direct products and strict unit $\mathbb{1}$ will usually be abbreviated as *allegory with some sharp products*. □

It is easy to see that every unit according to Def. 3.1.9 can serve as a strict unit for a choice of products, and that every strict unit for a choice of products is in fact a unit according to Def. 3.1.9.

The wide-spread prejudice that strictly associative choices of direct products would give rise to inconsistencies stems from the bad habit to write "$\pi_{\mathcal{A}\times\mathcal{B}}$" instead of "$\pi_{\mathcal{A},\mathcal{B}}$". With the latter, we can properly distinguish $\pi_{\mathcal{A}\times\mathcal{B},\mathcal{C}}$ and $\pi_{\mathcal{A},\mathcal{B}\times\mathcal{C}}$ and therefore do not obtain any inconsistencies.

Usually, however, we omit indices also for $\pi$ and $\rho$.

Given a choice of direct products, we may define *parallel composition* of relations: For two relations $Q : \mathcal{A} \leftrightarrow \mathcal{C}$ and $R : \mathcal{B} \leftrightarrow \mathcal{D}$, their *direct product* is relation from $\mathcal{A} \times \mathcal{B}$ to $\mathcal{C} \times \mathcal{D}$, defined as follows:

$$Q \times R := \pi ; Q ; \pi^{\smile} \sqcap \rho ; R ; \rho^{\smile}$$

In the context of a strictly associative choice of direct products, we shall write $R_1 \times \cdots \times R_n$ for the direct products of $n$ relations $R_1, \ldots, R_n$, and we usually assume projections $\pi_i$ for the $i$-th components of the involved direct products of objects. In the presence of a unit, the direct product of zero relations can, of course, only be the identity on the unit object.

The following fact holds even without sharpness:

**Lemma 3.1.13** [←79, 81, 82] If for two relations $R : \mathcal{A} \leftrightarrow \mathcal{C}$ and $S : \mathcal{B} \leftrightarrow \mathcal{D}$ in a locally co-complete allegory the products $\mathcal{A} \times \mathcal{B}$ and $\mathcal{C} \times \mathcal{D}$ exist, then we have:

$$\pi_{\mathcal{A},\mathcal{B}} ; R ; \mathbb{T}_{\mathcal{C},\mathcal{E}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \mathbb{T}_{\mathcal{D},\mathcal{E}} = (R \times S) ; \mathbb{T}_{\mathcal{C}\times\mathcal{D},\mathcal{E}}$$

**Proof**:

$$\begin{array}{lll}
 & \pi_{\mathcal{A},\mathcal{B}} ; R ; \mathbb{T}_{\mathcal{C},\mathcal{E}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \mathbb{T}_{\mathcal{D},\mathcal{E}} & \\
\sqsubseteq & (\pi_{\mathcal{A},\mathcal{B}} ; R \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \mathbb{T}_{\mathcal{D},\mathcal{E}} ; \mathbb{T}_{\mathcal{E},\mathcal{C}}) ; \mathbb{T}_{\mathcal{C},\mathcal{E}} & \text{modal rule} \\
\sqsubseteq & (\pi_{\mathcal{A},\mathcal{B}} ; R \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \mathbb{T}_{\mathcal{D},\mathcal{C}}) ; \mathbb{T}_{\mathcal{C},\mathcal{E}} & \\
= & (\pi_{\mathcal{A},\mathcal{B}} ; R \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \rho^{\smile}_{\mathcal{C},\mathcal{D}} ; \pi_{\mathcal{C},\mathcal{D}}) ; \mathbb{T}_{\mathcal{C},\mathcal{E}} & \\
= & (\pi_{\mathcal{A},\mathcal{B}} ; R ; \pi^{\smile}_{\mathcal{C},\mathcal{D}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \rho^{\smile}_{\mathcal{C},\mathcal{D}}) ; \pi_{\mathcal{C},\mathcal{D}} ; \mathbb{T}_{\mathcal{C},\mathcal{E}} & \pi_{\mathcal{C},\mathcal{D}} \text{ univalent} \\
\sqsubseteq & (\pi_{\mathcal{A},\mathcal{B}} ; R ; \pi^{\smile}_{\mathcal{C},\mathcal{D}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \rho^{\smile}_{\mathcal{C},\mathcal{D}}) ; \mathbb{T}_{\mathcal{C}\times\mathcal{D},\mathcal{E}} & \\
= & (R \times S) ; \mathbb{T}_{\mathcal{C}\times\mathcal{D},\mathcal{E}} & \\
= & (\pi_{\mathcal{A},\mathcal{B}} ; R ; \pi^{\smile}_{\mathcal{C},\mathcal{D}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \rho^{\smile}_{\mathcal{C},\mathcal{D}}) ; \mathbb{T}_{\mathcal{C}\times\mathcal{D},\mathcal{E}} & \\
\sqsubseteq & \pi_{\mathcal{A},\mathcal{B}} ; R ; \pi^{\smile}_{\mathcal{C},\mathcal{D}} ; \mathbb{T}_{\mathcal{C}\times\mathcal{D},\mathcal{E}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \rho^{\smile}_{\mathcal{C},\mathcal{D}} ; \mathbb{T}_{\mathcal{C}\times\mathcal{D},\mathcal{E}} & \\
\sqsubseteq & \pi_{\mathcal{A},\mathcal{B}} ; R ; \mathbb{T}_{\mathcal{C},\mathcal{E}} \sqcap \rho_{\mathcal{A},\mathcal{B}} ; S ; \mathbb{T}_{\mathcal{D},\mathcal{E}} &
\end{array}$$

□



## 3.2 Abstract Σ-Algebras and Relational Homomorphisms

In Sect. 2.3 we introduced conventional Σ-algebras, with sets as carriers and concrete total functions between these sets as interpretations of the function symbols. Such an approach has the advantage that it is more familiar to most readers, but it also has the disadvantage that it unduly constrains possible interpretations. We have seen that categories are the abstract version of settings with sets and total functions in-between, so we might define abstract Σ-algebras over categories with categorical products (see page 26).

However, since we are interested in *relational* morphisms between abstract Σ-algebras, we shall need an allegory setting as soon as we define morphisms. However, replacing a category in a definition of abstract Σ-algebras with an allegory is non-trivial, since we do not want to use the underlying category of the allegory for interpreting the signature, but the category of mappings *contained* in the chosen allegory.

In order to avoid superfluous technicalities for moving to and fro between these two levels, we base our definition of abstract Σ-algebras immediately on an allegory.

Instead of carrier sets, we then have arbitrary objects of the allegory, and instead of total functions between sets, we have mappings in the allegory. Furthermore, we have to use direct products and a unit for the domains of non-unary functions.

**Definition 3.2.1** [←30, 77, 90] Given a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and an allegory **D** with some sharp products, an *abstract Σ-algebra* $\mathcal{A}$ over **D** consists of the following items:

- for every sort $s : \mathcal{S}$, an object $s^{\mathcal{A}} \in Obj_{\mathbf{D}}$, such that for every function symbol $f \in \mathcal{F}$, the product $s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}$ exists, and
- for every function symbol $f \in \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$ a mapping $f^{\mathcal{A}} : s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}} \to t^{\mathcal{A}}$ in **D**. □

Since we use this definition to construct an allegory with abstract Σ-algebras as objects, the generality of discussing *abstract* Σ-algebras over allegories allows us to stack this construction at no cost at all, with possibly different signatures at every level, building for example graphs where the nodes and edges are hypergraphs.

The morphisms in allegories of Σ-algebras have to behave "essentially like relations", and so it is only natural that we consider a relational generalisation of conventional (functional) Σ-homomorphisms.

This is closely related to the field of data refinement, where usually unary homogeneous operations $f : s \to s$ over a signature with single sort $s$ are considered, and interpretations are allowed to be arbitrary relations, see for example the book by de Roever and Engelhardt [dRE98]. In that context, an "L-simulation" from $\mathcal{A}$ to $\mathcal{B}$ is a relation $\Phi : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{B}}$ satisfying the following inclusion:

$$\Phi^{\smile} ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{B}} ; \Phi^{\smile}$$

The name "L-simulation" is derived from the L-shape of the inclusion's left-hand side in the following sub-commuting diagram:

$$\begin{tikzcd} s^{\mathcal{B}} \arrow[r, "f^{\mathcal{B}}"] & s^{\mathcal{B}} \\ s^{\mathcal{A}} \arrow[u, "\Phi"] \arrow[r, "f^{\mathcal{A}}"'] \arrow[ur, phantom, "\sqsubseteq"] & s^{\mathcal{A}} \arrow[u, "\Phi"'] \end{tikzcd}$$

The corresponding inclusion for $n$-ary operations in a multi-sorted signature is then the following:

$$(\Phi_{s_1}^{\smile} \times \cdots \times \Phi_{s_n}^{\smile});f^{\mathcal{A}} \sqsubseteq f^{\mathcal{B}};\Phi_t^{\smile}$$

If we wanted to include relational structures in our considerations, we would have to use this as our relational homomorphism condition. However, relational structures with such a relational homomorphism concept only give rise to a category, but not to an allegory, since converse and meet do not preserve the above condition for interpretations $f^{\mathcal{A}}$ and $f^{\mathcal{B}}$ that are not mappings.

This explains why we had to exclude relational structures from our considerations and restrict ourselves to total algebras. (In the context of mappings as homomorphisms between relational structures, [Sch77] contains probably the first relation-algebraic considerations, see also [SS93, Chapt. 7]. Detailed discussions of different (functional) homomorphism conditions for relational structures may be found in [WB98]. We are now dealing with *relational* homomorphisms between *functional* structures.)

So we always know that $f^{\mathcal{A}}$ and $f^{\mathcal{B}}$ are mappings, and then Lemma A.1.2.iii) allows us to move them to the respective other sides of the inclusion, yielding an equivalent formulation that does not contain converse; this is the inclusion we are going to use as our relational homomorphism condition:

**Definition 3.2.2** [←22] Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, an allegory $\mathbf{D}$, and two abstract $\Sigma$-algebras $\mathcal{A}$ and $\mathcal{B}$ over $\mathbf{D}$ be given.

A *$\Sigma$-compatible family of relations from $\mathcal{A}$ to $\mathcal{B}$* is a $\mathcal{S}$-indexed family of relations $\Phi = (\Phi_s)_{s:\mathcal{S}}$ such that $\Phi_s : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{B}}$ for every sort $s : \mathcal{S}$.

A *relational $\Sigma$-algebra homomorphism from $\mathcal{A}$ to $\mathcal{B}$* is a $\Sigma$-compatible family of relations from $\mathcal{A}$ to $\mathcal{B}$ such that for every function symbol $f \in \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$, the following inclusion holds:

$$(\Phi_{s_1} \times \cdots \times \Phi_{s_n});f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}};\Phi_t \ . \qquad \square$$

In the allegory $\mathbf{D}$, this gives rise to the following sub-commuting diagrams, one for a constant $c : t$, one for a unary function symbol $g : s \to t$, and one for an $n$-ary function symbol $f : s_1 \times \cdots \times s_n \to t$ (arranged in a different way than above):

$$\begin{tikzcd} \mathbb{1} \arrow[r, "c^{\mathcal{A}}"] \arrow[d, "\mathbb{I}_{\mathbb{1}}"'] \arrow[dr, phantom, "\sqsubseteq"] & t^{\mathcal{A}} \arrow[d, "\Phi_t"] \\ \mathbb{1} \arrow[r, "c^{\mathcal{B}}"'] & t^{\mathcal{B}} \end{tikzcd} \qquad \begin{tikzcd} s^{\mathcal{A}} \arrow[r, "g^{\mathcal{A}}"] \arrow[d, "\Phi_s"'] \arrow[dr, phantom, "\sqsubseteq"] & t^{\mathcal{A}} \arrow[d, "\Phi_t"] \\ s^{\mathcal{B}} \arrow[r, "g^{\mathcal{B}}"'] & t^{\mathcal{B}} \end{tikzcd} \qquad \begin{tikzcd} s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}} \arrow[r, "f^{\mathcal{A}}"] \arrow[d, "\Phi_{s_1} \times \cdots \times \Phi_{s_n}"'] \arrow[dr, phantom, "\sqsubseteq"] & t^{\mathcal{A}} \arrow[d, "\Phi_t"] \\ s_1^{\mathcal{B}} \times \cdots \times s_n^{\mathcal{B}} \arrow[r, "f^{\mathcal{B}}"'] & t^{\mathcal{B}} \end{tikzcd}$$



Now we first justify the "homomorphism" part of the name of "relational $\Sigma$-algebra homomorphisms", that is, we show that we really obtain a category:

**Proposition 3.2.3** Given an allegory $\mathbf{D}$ with some sharp products and a signature $\Sigma$, relational $\Sigma$-algebra homomorphisms form a category, where composition and identities are defined component-wise.

**Proof**: First we show that $\mathbb{I}_{\mathcal{A}}$ is a relational $\Sigma$-algebra homomorphism:

- $(\mathbb{I}_{\mathcal{A}})_s = \mathbb{I}_{s^{\mathcal{A}}} : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{A}}$, and
- for all $f \in \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$ we have

$$\begin{array}{rcll} ((\mathbb{I}_{\mathcal{A}})_{s_1} \times \cdots \times (\mathbb{I}_{\mathcal{A}})_{s_n});f^{\mathcal{A}} & = & (\mathbb{I}_{s_1^{\mathcal{A}}} \times \cdots \times \mathbb{I}_{s_n^{\mathcal{A}}});f^{\mathcal{A}} & \\ & = & \mathbb{I}_{s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}};f^{\mathcal{A}} & \text{Def. direct product} \\ & = & f^{\mathcal{A}};\mathbb{I}_{t^{\mathcal{A}}} & \\ & = & f^{\mathcal{A}};(\mathbb{I}_{\mathcal{A}})_t \ . & \end{array}$$

Now we show well-definedness of composition: Let $\Psi : \mathcal{A} \to \mathcal{B}$ and $\Psi : \mathcal{B} \to \mathcal{B}$ be relational $\Sigma$-algebra homomorphisms, and $\Xi := \Phi;\Psi$, then:

- $\Xi_s = \Phi_s;\Psi_s : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{C}}$, and
- for all $f \in \mathcal{F}$ with $f : s_1 \times \cdots \times s_n \to t$ we have

$$\begin{array}{cll} & (\Xi_{s_1} \times \cdots \times \Xi_{s_n});f^{\mathcal{C}} & \\ = & (\Phi_{s_1};\Psi_{s_1} \times \cdots \times \Phi_{s_n};\Psi_{s_n});f^{\mathcal{C}} & \\ \sqsubseteq & (\Phi_{s_1} \times \cdots \times \Phi_{s_n});(\Psi_{s_1} \times \cdots \times \Psi_{s_n});f^{\mathcal{C}} & \text{sharp products} \\ \sqsubseteq & (\Phi_{s_1} \times \cdots \times \Phi_{s_n});f^{\mathcal{B}};\Psi_t & \Psi \text{ relational homomorphism} \\ \sqsubseteq & f^{\mathcal{A}};\Phi_t;\Psi_t & \Phi \text{ relational homomorphism} \\ = & f^{\mathcal{A}};\Xi_t \ . & \end{array}$$

Associativity of composition and the identity laws follow via the component-wise definitions. $\square$

In the same way, we can lift converse, meet and universal relations from the underlying allegory to relational homomorphisms:

**Definition 3.2.4** Given an allegory $\mathbf{D}$ with some sharp products and a signature $\Sigma$, we define the following operations on relational $\Sigma$-algebra homomorphisms:

(i) If $\Phi : \mathcal{A} \to \mathcal{B}$ is a relational homomorphism, then the *converse* of $\Phi = (\Phi_s)_{s:\mathcal{S}}$ is $\Phi^{\smile} := (\Phi_s^{\smile})_{s:\mathcal{S}}$.

(ii) If $\Phi, \Psi : \mathcal{A} \to \mathcal{B}$ are relational homomorphisms, then their *meet* is defined component-wise: $(\Phi \sqcap \Psi)_s := \Phi_s \sqcap \Psi_s$

(iii) $\mathbb{T}_{\mathcal{A},\mathcal{B}} := (\mathbb{T}_{s^{\mathcal{A}},s^{\mathcal{B}}})_{s:\mathcal{S}}$, if the universal relations $\mathbb{T}_{s^{\mathcal{A}},s^{\mathcal{B}}}$ all exist.

**Proof of well-definedness**:

$$
\begin{array}{lll}
\text{(i)} & ((\Phi^{\smile})_{s_1} \times \cdots \times (\Phi^{\smile})_{s_n}) ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{B}} ; (\Phi^{\smile})_t & \\
\Leftrightarrow & (\Phi^{\smile}_{s_1} \times \cdots \times \Phi^{\smile}_{s_n}) ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{B}} ; \Phi^{\smile}_t & \\
\Leftrightarrow & f^{\mathcal{B}\smile} ; (\Phi^{\smile}_{s_1} \times \cdots \times \Phi^{\smile}_{s_n}) \sqsubseteq \Phi^{\smile}_t ; f^{\mathcal{A}\smile} & f^{\mathcal{A}},\ f^{\mathcal{B}} \text{ mappings, Lemma A.1.2.iii)} \\
\Leftrightarrow & (\Phi_{s_1} \times \cdots \times \Phi_{s_n}) ; f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}} ; \Phi_t &
\end{array}
$$

$$
\begin{array}{lll}
\text{(ii)} & ((\Phi \sqcap \Psi)_{s_1} \times \cdots \times (\Phi \sqcap \Psi)_{s_n}) ; f^{\mathcal{B}} & \\
= & ((\Phi_{s_1} \sqcap \Psi_{s_1}) \times \cdots \times (\Phi_{s_n} \sqcap \Psi_{s_n})) ; f^{\mathcal{B}} & \\
= & ((\Phi_{s_1} \times \cdots \times \Phi_{s_n}) \sqcap (\Psi_{s_1} \times \cdots \times \Psi_{s_n})) ; f^{\mathcal{B}} & \text{Lemma A.2.3} \\
\sqsubseteq & (\Phi_{s_1} \times \cdots \times \Phi_{s_n}) ; f^{\mathcal{B}} \sqcap (\Psi_{s_1} \times \cdots \times \Psi_{s_n}) ; f^{\mathcal{B}} & \\
\sqsubseteq & f^{\mathcal{A}} ; \Phi_t \sqcap f^{\mathcal{A}} ; \Psi_t & \\
= & f^{\mathcal{A}} ; (\Phi_t \sqcap \Psi_t) & f^{\mathcal{A}} \text{ univalent} \\
= & f^{\mathcal{A}} ; (\Phi \sqcap \Psi)_t &
\end{array}
$$

(iii) $(\mathbb{T}_{s_1^{\mathcal{A}}, s_1^{\mathcal{B}}} \times \cdots \times \mathbb{T}_{s_n^{\mathcal{A}}, s_n^{\mathcal{B}}}) ; f^{\mathcal{B}} \sqsubseteq \mathbb{T}_{s_1^{\mathcal{A}} \times \cdots \times s_n^{\mathcal{A}}, t^{\mathcal{B}}} = f^{\mathcal{A}} ; \mathbb{T}_{t^{\mathcal{A}}, t^{\mathcal{B}}}$, since $f^{\mathcal{A}}$ is total. □

Inclusion between homomorphisms may therefore be defined as usual via

$$\Phi \sqsubseteq \Psi \quad :\Leftrightarrow \quad \Phi \sqcap \Psi = \Phi \ ,$$

and this then is equivalent to the component-wise definition of inclusion, because of the component-wise definition of meet.

**Proposition 3.2.5** [←88] Given a locally co-complete allegory **D** with some sharp products, a signature $\Sigma$, and a set $\mathcal{R}$ of relational $\Sigma$-algebra homomorphisms such that for every $\Phi \in \mathcal{R}$ we have $\Phi : \mathcal{A} \to \mathcal{B}$, then the meet over the set $\mathcal{R}$ exists and can be seen as defined component-wise:

$$(\textstyle\bigsqcap \mathcal{R})_s = \bigsqcap \{\Phi : \mathcal{R} \bullet \Phi_s\}$$

**Proof**: We show that the component-wise definition yields a well-defined relational homomorphism; by the component-wise definition of inclusion this implies that this is in fact the meet.

$$
\begin{array}{lll}
 & ((\bigsqcap \mathcal{R})_{s_1} \times \cdots \times (\bigsqcap \mathcal{R})_{s_n}) ; f^{\mathcal{B}} & \\
= & ((\bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_{s_1}\}) \times \cdots \times (\bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_{s_n}\})) ; f^{\mathcal{B}} & \\
= & (\bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_{s_1} \times \cdots \times \Phi_{s_n}\}) ; f^{\mathcal{B}} & \text{Lemma A.2.3} \\
\sqsubseteq & \bigsqcap\{\Phi : \mathcal{R} \bullet (\Phi_{s_1} \times \cdots \times \Phi_{s_n}) ; f^{\mathcal{B}}\} & \mathbf{D} \text{ locally co-complete} \\
\sqsubseteq & \bigsqcap\{\Phi : \mathcal{R} \bullet f^{\mathcal{A}} ; \Phi_t\} & \\
= & f^{\mathcal{A}} ; (\bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_t\}) & f^{\mathcal{A}} \text{ mapping, Lemma A.1.5} \\
= & f^{\mathcal{A}} ; (\bigsqcap \mathcal{R})_t &
\end{array}
$$

□

Given the closedness of relational $\Sigma$-algebra homomorphisms under converse and arbitrary meets, properties of relations for these operations are inherited by relational $\Sigma$-algebra homomorphisms because of the component-wise definition, so we immediately see that abstract $\Sigma$-algebras with relational homomorphisms form a locally co-complete allegory, which justifies the "relational" part of the name:



**Theorem 3.2.6 (Allegories of Σ-Algebras)** Abstract $\Sigma$-algebras over an allegory **D** with some sharp products together with relational $\Sigma$-algebra homomorphisms form an allegory, denoted $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$.

If **D** is locally co-complete, then so is $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$. □

This is an important first result; it also provides useful examples of allegories that are not distributive, as we shall see in Sect. 4.2.

We may observe a few simple facts:

- If **D** contains a unit $\mathbb{1}$, then $\mathbb{1}_\Sigma$ with $s^{\mathbb{1}_\Sigma} = \mathbb{1}$ for all sorts $s$ and $f^{\mathbb{1}_\Sigma} = \mathbb{I}_{\mathbb{1}}$ for all function symbols $f$ is an abstract $\Sigma$-algebra, and a unit in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$.
- If **D** contains an initial object $\emptyset$, and $\Sigma$ contains no constants, then $\mathbb{O}_\Sigma$ with $s^{\mathbb{O}_\Sigma} = \emptyset$ for all sorts $s$ and $f^{\mathbb{O}_\Sigma} = \mathbb{I}_\emptyset$ for all function symbols $f$ is an abstract $\Sigma$-algebra, and an initial object in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$.

Conventional $\Sigma$-algebra homomorphisms are just mappings between concrete $\Sigma$-algebras — we denote the allegory of sets and concrete relations with *Rel*:

**Proposition 3.2.7** [←79] For every signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, conventional $\Sigma$-algebra homomorphism are the mappings in $\Sigma\text{-}\mathsf{Alg}_{Rel}$.

**Proof**: A conventional $\Sigma$-algebra homomorphism between $\mathcal{A}$ and $\mathcal{B}$ is a family $(\Phi_s)_{s:\mathcal{S}}$ such that $\Phi_s$ is a mapping from $s^{\mathcal{A}}$ to $s^{\mathcal{B}}$ for every sort $s : \mathcal{S}$, and for every function symbol $f : s_1 \times \cdots \times s_n \to t$, the equation

$$(\Phi_{s_1} \times \cdots \times \Phi_{s_n}) ; f^{\mathcal{B}} = f^{\mathcal{A}} ; \Phi_t$$

holds in *Rel*.

Since this equation implies the relational homomorphism condition of Def. 3.2.1, every conventional $\Sigma$-algebra homomorphism is a relational homomorphism, and it is a mapping because of the component-wise definitions of converse, composition, identities and inclusion.

In the same way, these component-wise definitions imply that if $\Phi$ is a mapping in $\Sigma\text{-}\mathsf{Alg}_{Rel}$, then $\Phi_s$ is a mapping in *Rel* for every sort $s$. Since then all $\Phi_{s_i}$ are mappings, $\Phi_{s_1} \times \cdots \times \Phi_{s_n}$ is a mapping, too, and the homomorphism condition

$$(\Phi_{s_1} \times \cdots \times \Phi_{s_n}) ; f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}} ; \Phi_t$$

turns into an equality since both sides of the inclusion are mappings. □

## 3.3 Constructions in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$

We now show how a few standard constructions on algebras can still be performed in our abstract relational setting.

Essentially, these are well-known facts from universal algebra. There, however, carrier sets are not allowed to be empty. Therefore we spell out the proofs and thus show that this generalisation does not influence the results.

## Subobjects

A partial identity on an object $\mathcal{A}$ determines a substructure of $\mathcal{A}$. In the set-theoretic setting of Chapter 2, a subalgebra immediately is as an algebra in its on right. In the category-theoretic setting of this chapter, however, there need not exist an object corresponding to such a substructure — for example in one-object allegories.

Therefore, one always has to consider whether a subobject induced by a partial identity exists. In the language of Freyd and Scedrov, the question is whether a "coreflexive is split" or not. We stick to a more intuitive language, and define:

**Definition 3.3.1** In an allegory $\mathbf{D}$, let a partial identity $q : \mathsf{PId}\,\mathcal{A}$ be given. A *subobject for* $q$ is a pair $(\mathcal{S}, \lambda)$ consisting of an object $\mathcal{S}$ and an injective mapping $\lambda : \mathcal{S} \to \mathcal{A}$ such that $\mathsf{ran}\,\lambda = q$.

The allegory $\mathbf{D}$ *has subobjects* iff for every partial identity $q : \mathsf{PId}\,\mathcal{A}$ there is a subobject. □

**Theorem 3.3.2** Given a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and an allegory $\mathbf{D}$. Let $q : \mathsf{PId}\,\mathcal{A}$ be a partial identity on an object $\mathcal{A}$ in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$. If for every sort $s \in \mathcal{S}$, there is a subobject for $q_s$ in $\mathbf{D}$, then there is a subobject for $q$ in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$.

Therefore, if $\mathbf{D}$ has subobjects, then $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ has subobjects, too.

**Proof**: For every sort $s \in \mathcal{S}$, the relation $q_s$ is a partial identity on $\mathcal{A}_s$ and we may choose a subobject $(\mathcal{S}_s, \lambda_s)$ for $q_s$ in $\mathbf{D}$. Then we define $\mathcal{S}$ as follows:

- for every sort $s \in \mathcal{S}$, the carrier is $\mathcal{S}_s$;
- for every function symbol $f : s_1 \times \cdots \times s_n \to t$ we define

$$f^{\mathcal{S}} := (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} ; \lambda_t^{\smile} \ .$$

Then $\mathcal{S}$ is a $\Sigma$-algebra:

- Univalence of $f^{\mathcal{S}}$:

$$\begin{array}{rcll}
f^{\mathcal{S}\smile} ; f^{\mathcal{S}} & = & \lambda_t ; f^{\mathcal{A}\smile} ; (\lambda_{s_1}^{\smile} \times \cdots \times \lambda_{s_n}^{\smile}) ; (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} ; \lambda_t^{\smile} & \\
& \sqsubseteq & \lambda_t ; f^{\mathcal{A}\smile} ; (\lambda_{s_1}^{\smile} ; \lambda_{s_1} \times \cdots \times \lambda_{s_n}^{\smile} ; \lambda_{s_n}) ; f^{\mathcal{A}} ; \lambda_t^{\smile} & \\
& = & \lambda_t ; f^{\mathcal{A}\smile} ; (q_{s_1} \times \cdots \times q_{s_n}) ; f^{\mathcal{A}} ; \lambda_t^{\smile} & \\
& \sqsubseteq & \lambda_t ; f^{\mathcal{A}\smile} ; f^{\mathcal{A}} ; \lambda_t^{\smile} & q_{s_i} \sqsubseteq \mathbb{I} \\
& \sqsubseteq & \lambda_t ; \lambda_t^{\smile} & f^{\mathcal{A}} \text{ univalent} \\
& \sqsubseteq & \mathbb{I} & \lambda_t \text{ univalent}
\end{array}$$



- Totality of $f^{\mathcal{S}}$:

$$\begin{array}{rcll}
f^{\mathcal{S}} ; \mathbb{T} & = & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} ; \lambda_t^{\smile} ; \mathbb{T} & \\
& = & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} ; q_t ; \mathbb{T} & \\
& \sqsupseteq & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; (q_{s_1} \times \cdots \times q_{s_n}) ; f^{\mathcal{A}} ; \mathbb{T} & q \text{ relational homom.} \\
& = & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; (q_{s_1} \times \cdots \times q_{s_n}) ; \mathbb{T} & f^{\mathcal{A}} \text{ total} \\
& \sqsupseteq & (\lambda_{s_1} ; q_{s_1} \times \cdots \times \lambda_{s_n} ; q_{s_n}) ; \mathbb{T} & \text{sharp products} \\
& = & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; \mathbb{T} & \\
& \sqsupseteq & \pi_1 ; \lambda_{s_1} ; \mathbb{T} \sqcap \cdots \sqcap \pi_n ; \lambda_{s_n} ; \mathbb{T} & \text{Lemma 3.1.13} \\
& = & \mathbb{T} & \pi_i,\ \lambda_{s_i} \text{ total}
\end{array}$$

In addition, $\lambda := (\lambda_s)_{s:\mathcal{S}}$ is a relational homomorphism since:

$$\begin{array}{rll}
& (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} & \\
= & (\lambda_{s_1} ; q_{s_1} \times \cdots \times \lambda_{s_n} ; q_{s_n}) ; f^{\mathcal{A}} & \\
\sqsubseteq & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; (q_{s_1} \times \cdots \times q_{s_n}) ; f^{\mathcal{A}} & \text{sharp products} \\
\sqsubseteq & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} ; q_t & q \text{ relational homomorphism} \\
= & (\lambda_{s_1} \times \cdots \times \lambda_{s_n}) ; f^{\mathcal{A}} ; \lambda_t^{\smile} ; \lambda_t & \lambda_t \text{ univalent} \\
= & f^{\mathcal{S}} ; \lambda_t &
\end{array}$$

In all these arguments, the $n$-ary products of relations have to be replaced by $\mathbb{I}_{\mathbb{1}}$ for zero-ary function symbols, but this does not change the arguments.

Totality, univalence and injectivity of $\lambda$ are obvious from its definition; we also have

$$\mathsf{ran}\,\lambda = (\mathsf{ran}\,\lambda_s)_{s:\mathcal{S}} = (q_s)_{s:\mathcal{S}} = q$$

□

## Direct Quotients

An injective mapping $F$ always can be seen as establishing a subobject relation between its source and target objects: the source is a subobject of the target for the partial identity $F^{\smile} ; F$.

Dually, for a surjective mapping $G : \mathcal{A} \leftrightarrow \mathcal{B}$, we obtain an equivalence relation $G ; G^{\smile}$, and $G$ acts as the *projection* of a *quotient* of $\mathcal{A}$ by this equivalence relation.

In the context of $\Sigma$-algebras, quotients are taken by congruences, so let us first present the conventional definition of congruences, adapted to our abstract setting:

**Definition 3.3.3** Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ be given. For an abstract $\Sigma$-algebra $\mathcal{A}$, a $\Sigma$-*congruence* is a family $(\Xi_s)_{s:\mathcal{S}}$ of equivalence relations $\Xi_s : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{A}}$ such that for all function symbols $f : s \to t$ in $\mathcal{F}$ the following holds:

$$\Xi_s ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{A}} ; \Xi_t$$

□

Just as, according to Proposition 3.2.7, conventional $\Sigma$-algebra homomorphisms turn into simple mappings in our setting, conventional $\Sigma$-congruences turn into simple equivalence relations:

**Theorem 3.3.4** Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ be given. $\Xi : \mathcal{A} \leftrightarrow \mathcal{A}$ is an equivalence relation in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ iff $\Xi$ is a $\Sigma$-congruence on the $\Sigma$-algebra $\mathcal{A}$.

**Proof**: Assume $\Xi$ is an equivalence relation in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$. Then for every sort $s : \mathcal{S}$, the component $\Xi_s$ is an equivalence relation since reflexivity, transitivity and antisymmetry of $\Xi$ all propagate to the components via component-wise definitions of the involved operators, and for every function symbol $f : s \to t$ in $\mathcal{F}$ the congruence inclusion $\Xi_s ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{A}} ; \Xi_t$ holds since $\Xi$ is a relational homomorphism.

Now assume $(\Xi_s)_{s:\mathcal{S}}$ is a $\Sigma$-congruence on $\mathcal{A}$. Then $\Xi$ is a relational homomorphism because of the congruence condition, and $\Xi$ is an equivalence relation in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ since reflexivity, transitivity and antisymmetry all follow from the component-wise definitions of the involved operators. □

**Definition 3.3.5** A *direct quotient* for an equivalence relation $\Xi : \mathcal{A} \leftrightarrow \mathcal{A}$ is a pair $(\mathcal{Q}, \theta)$ consisting of an object $\mathcal{Q}$ together with a surjective mapping $\theta : \mathcal{A} \to \mathcal{Q}$ such that $\theta ; \theta^{\smile} = \Xi$.

An allegory $\mathbf{D}$ *has direct quotients* iff for every equivalence relation there is a direct quotient. □

**Theorem 3.3.6** Let an allegory $\mathbf{D}$ and a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ be given, and let $\Xi : \mathcal{A} \leftrightarrow \mathcal{A}$ be an equivalence relation in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$. If for every sort $s \in \mathcal{S}$, there is a direct quotient $(\mathcal{Q}_s, \theta_s)$ for $\Xi_s$ in $\mathbf{D}$, then there is also a direct quotient $(\mathcal{Q}, \theta)$ for $\Xi$ in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$.

Therefore, if $\mathbf{D}$ has direct quotients, then $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ has direct quotients, too.

**Proof**: For every sort $s \in \mathcal{S}$, the relation $\Xi_s$ is an equivalence relation in $\mathbf{D}$, and according to the assumption we may choose a direct quotient $(\mathcal{Q}_s, \theta_s)$ for $\Xi_s$ in $\mathbf{D}$.

Then we define $\mathcal{Q}$ as follows:

- for every sort $s \in \mathcal{S}$, the carrier is $\mathcal{Q}_s$;
- for every function symbol $f : s_1 \times \cdots \times s_n \to t$ we define

$$f^{\mathcal{Q}} := (\theta^{\smile}_{s_1} \times \cdots \times \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \theta_t \ .$$

Then $\mathcal{Q}$ is a $\Sigma$-algebra:

- $f^{\mathcal{Q}}$ is univalent since

$$\begin{array}{rcll}
f^{\mathcal{Q}\smile} ; f^{\mathcal{Q}} & = & \theta^{\smile}_t ; f^{\mathcal{A}\smile} ; (\theta_{s_1} \times \cdots \times \theta_{s_n}) ; (\theta^{\smile}_{s_1} \times \cdots \times \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \theta_t & \\
& \sqsubseteq & \theta^{\smile}_t ; f^{\mathcal{A}\smile} ; (\theta_{s_1} ; \theta^{\smile}_{s_1} \times \cdots \times \theta_{s_n} ; \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \theta_t & \text{product properties} \\
& = & \theta^{\smile}_t ; f^{\mathcal{A}\smile} ; (\Xi_{s_1} \times \cdots \times \Xi_{s_n}) ; f^{\mathcal{A}} ; \theta_t & \\
& \sqsubseteq & \theta^{\smile}_t ; f^{\mathcal{A}\smile} ; f^{\mathcal{A}} ; \Xi_t ; \theta_t & \Xi \text{ rel. hom.} \\
& = & \theta^{\smile}_t ; f^{\mathcal{A}\smile} ; f^{\mathcal{A}} ; \theta_t & \\
& \sqsubseteq & \theta^{\smile}_t ; \theta_t & f^{\mathcal{A}} \text{ univalent} \\
& \sqsubseteq & \mathbb{I} & \theta_t \text{ univalent}
\end{array}$$



- if $f$ is nullary, then $f^{\mathcal{Q}}$ is total since

$$f^{\mathcal{Q}} ; \mathbb{T} = \mathbb{I}_{\mathbb{1}} ; f^{\mathcal{A}} ; \theta_t ; \mathbb{T} = \mathbb{I}_{\mathbb{1}} ; f^{\mathcal{A}} ; \mathbb{T} = \mathbb{I}_{\mathbb{1}} ; \mathbb{T} = \mathbb{T} \ ;$$

otherwise, $f^{\mathcal{Q}}$ is total since

$$\begin{array}{rcll}
f^{\mathcal{Q}} ; \mathbb{T} & = & (\theta^{\smile}_{s_1} \times \cdots \times \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \theta_t ; \mathbb{T} & \\
& = & (\theta^{\smile}_{s_1} \times \cdots \times \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \mathbb{T} & \theta_t \text{ total} \\
& = & (\theta^{\smile}_{s_1} \times \cdots \times \theta^{\smile}_{s_n}) ; \mathbb{T} & f^{\mathcal{A}} \text{ total} \\
& = & \pi_1 ; \theta^{\smile}_{s_1} ; \mathbb{T} \sqcap \ldots \sqcap \pi_n ; \theta^{\smile}_{s_n} ; \mathbb{T} & \text{Lemma 3.1.13} \\
& = & \pi_1 ; \mathbb{T} \sqcap \ldots \sqcap \pi_n ; \mathbb{T} & \theta_{s_i} \text{ surjective} \\
& = & \mathbb{T} & \pi_i \text{ total.}
\end{array}$$

Now let $\theta := (\theta_s)_{s:\mathcal{S}}$. Then $\theta$ is total, univalent and surjective because its components are, and it is a relational homomorphism from $\mathcal{Q}$ to $\mathcal{A}$:

$$\begin{array}{rll}
& (\theta_{s_1} \times \cdots \times \theta_{s_n}) ; f^{\mathcal{Q}} & \\
= & (\theta_{s_1} \times \cdots \times \theta_{s_n}) ; (\theta^{\smile}_{s_1} \times \cdots \times \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \theta_t & \\
\sqsubseteq & (\theta_{s_1} ; \theta^{\smile}_{s_1} \times \cdots \times \theta_{s_n} ; \theta^{\smile}_{s_n}) ; f^{\mathcal{A}} ; \theta_t & \text{product properties} \\
= & (\Xi_{s_1} \times \cdots \times \Xi_{s_n}) ; f^{\mathcal{A}} ; \theta_t & \\
\sqsubseteq & f^{\mathcal{A}} ; \Xi_t ; \theta_t & \Xi \text{ relational homomorphism} \\
= & f^{\mathcal{A}} ; \theta_t &
\end{array}$$

The equation $\theta ; \theta^{\smile} = \Xi$ follows from the component-wise definition of $\theta$, too. □

## Direct Products

Being careful not to unnecessarily assume existence of all direct products, we see that for the existence of the direct product of two algebras, the direct products of the corresponding carriers in the underlying allegory have to exist.

It is therefore possible that in some suitably chosen suballegory of $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ all products exist, although not all products exist in the corresponding suballegory of $\mathbf{D}$. In such a suballegory of $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$, certain objects of $\mathbf{D}$ would occur as carriers only for certain sorts, and other objects for other sorts. Since such a suballegory might be induced by laws, this could give rise to interesting constellations.

**Theorem 3.3.7** Let an allegory $\mathbf{D}$, a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, and two objects $\mathcal{A}$ and $\mathcal{B}$ in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ be given. If for every sort $s \in \mathcal{S}$, there is a direct product for $s^{\mathcal{A}}$ and $s^{\mathcal{B}}$ in $\mathbf{D}$, then there is a direct product for $\mathcal{A}$ and $\mathcal{B}$ in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$.

Therefore, if $\mathbf{D}$ has direct products, then $\Sigma{-}\mathsf{Alg}_{\mathbf{D}}$ has direct products, too.

**Proof**: For every sort $s : \mathcal{S}$, let us assume a direct product $(\mathcal{P}_s, \pi_s, \rho_s)$ of $s^{\mathcal{A}}$ and $s^{\mathcal{B}}$. Then we let $\mathcal{P}$ be defined by these $\mathcal{P}_s$ as carriers and by defining for every function symbol $f : s_1 \times \cdots \times s_n \to t$ the mapping

$$f^{\mathcal{P}} := (\pi_{s_1} \times \cdots \times \pi_{s_n}) ; f^{\mathcal{A}} ; \pi^{\smile}_t \sqcap (\rho_{s_1} \times \cdots \times \rho_{s_n}) ; f^{\mathcal{B}} ; \rho^{\smile}_t$$

(For zero-ary function symbols $c : t$, this degenerates to $c^{\mathcal{B}} := c^{\mathcal{A}} ; \pi^{\smile}_t \sqcap c^{\mathcal{B}} ; \rho^{\smile}_t$, but this does not affect the following arguments.)

For showing totality of $f^{\mathcal{P}}$, we use essentially the argument of Lemma 3.1.13:

$$\begin{array}{lll}
 & f^{\mathcal{P}};\mathbb{T}_{t^{\mathcal{P}},\mathcal{X}} & \\
= & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}};\pi_t^{\smile}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile});\mathbb{T}_{t^{\mathcal{A}}\times t^{\mathcal{B}},\mathcal{X}} & \\
= & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}};\pi_t^{\smile}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile});\pi_t;\mathbb{T}_{t^{\mathcal{A}}\times t^{\mathcal{B}},\mathcal{X}} & \pi_t \text{ total}\\
= & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile};\pi_t);\mathbb{T}_{t^{\mathcal{A}},\mathcal{X}} & \pi_t \text{ univalent}\\
= & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\mathbb{T}_{t^{\mathcal{B}},t^{\mathcal{A}}});\mathbb{T}_{t^{\mathcal{A}},\mathcal{X}} & \\
\sqsupseteq & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\mathbb{T}_{t^{\mathcal{B}},\mathcal{X}};\mathbb{T}_{\mathcal{X},t^{\mathcal{A}}});\mathbb{T}_{t^{\mathcal{A}},\mathcal{X}} & \\
\sqsupseteq & (\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}};\mathbb{T}_{t^{\mathcal{A}},\mathcal{X}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\mathbb{T}_{t^{\mathcal{B}},\mathcal{X}} & \text{modal rule}\\
= & (\pi_{s_1}\times\cdots\times\pi_{s_n});\mathbb{T}_{s_1^{\mathcal{A}}\times\cdots\times s_n^{\mathcal{A}},\mathcal{X}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});\mathbb{T}_{s_1^{\mathcal{B}}\times\cdots\times s_n^{\mathcal{B}},\mathcal{X}} & f^{\mathcal{A}},\ f^{\mathcal{B}} \text{ total}\\
= & \pi_1;\pi_{s_1};\mathbb{T}_{s_1^{\mathcal{A}},\mathcal{X}}\sqcap\ldots\sqcap\pi_n;\pi_{s_n};\mathbb{T}_{s_n^{\mathcal{A}},\mathcal{X}}\sqcap\pi_1;\rho_{s_1};\mathbb{T}_{s_1^{\mathcal{B}},\mathcal{X}}\sqcap\ldots\sqcap\pi_n;\rho_{s_n};\mathbb{T}_{s_n^{\mathcal{B}},\mathcal{X}} & 3.1.13\\
= & \mathbb{T}_{s_1^{\mathcal{P}}\times\cdots\times s_n^{\mathcal{P}},\mathcal{X}} & \pi_i,\ \pi_{s_i} \text{ total}
\end{array}$$

Univalence:

$$\begin{array}{lll}
 & (f^{\mathcal{P}})^{\smile};f^{\mathcal{P}} & \\
= & (\pi_t;(f^{\mathcal{A}})^{\smile};(\pi_{s_1}^{\smile}\times\cdots\times\pi_{s_n}^{\smile})\sqcap\rho_t;(f^{\mathcal{B}})^{\smile};(\rho_{s_1}^{\smile}\times\cdots\times\rho_{s_n}^{\smile})); & \\
 & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}};\pi_t^{\smile}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile}) & \\
\sqsubseteq & \pi_t;(f^{\mathcal{A}})^{\smile};(\pi_{s_1}^{\smile}\times\cdots\times\pi_{s_n}^{\smile});(\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}};\pi_t^{\smile}\sqcap & \\
 & \rho_t;(f^{\mathcal{B}})^{\smile};(\rho_{s_1}^{\smile}\times\cdots\times\rho_{s_n}^{\smile});(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile} & \\
\sqsubseteq & \pi_t;(f^{\mathcal{A}})^{\smile};(\pi_{s_1}^{\smile};\pi_{s_1}\times\cdots\times\pi_{s_n}^{\smile};\pi_{s_n});f^{\mathcal{A}};\pi_t^{\smile}\sqcap & \\
 & \rho_t;(f^{\mathcal{B}})^{\smile};(\rho_{s_1}^{\smile};\rho_{s_1}\times\cdots\times\rho_{s_n}^{\smile};\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile} & \text{dir. prod. properties}\\
\sqsubseteq & \pi_t;(f^{\mathcal{A}})^{\smile};f^{\mathcal{A}};\pi_t^{\smile}\sqcap\rho_t;(f^{\mathcal{B}})^{\smile};f^{\mathcal{B}};\rho_t^{\smile} & \pi_{s_i},\ \rho_{s_i} \text{ univalent}\\
\sqsubseteq & \pi_t;\pi_t^{\smile}\sqcap\rho_t;\rho_t & f^{\mathcal{A}},\ f^{\mathcal{B}} \text{ univalent}\\
= & \mathbb{I}_{t^{\mathcal{P}}} &
\end{array}$$

Furthermore, $\pi_{\mathcal{P}} := (\pi_s)_{s:\mathcal{S}}$ and $\rho_{\mathcal{P}} := (\rho_s)_{s:\mathcal{S}}$ are by definition total, univalent and surjective, and they are also relational homomorphisms as we show only for $\pi_{\mathcal{P}}$:

$$\begin{array}{lll}
 & (\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}} & \\
= & (\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap\mathbb{T}_{s^{\mathcal{P}},t^{\mathcal{A}}} & \\
= & (\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});\mathbb{T}_{s^{\mathcal{B}},t^{\mathcal{A}}} & \rho_{s_i} \text{ total, Lemma 3.1.13}\\
= & (\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\mathbb{T}_{t^{\mathcal{B}},t^{\mathcal{A}}} & f^{\mathcal{B}} \text{ total}\\
= & (\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t^{\smile};\pi_t & \\
= & ((\pi_{s_1}\times\cdots\times\pi_{s_n});f^{\mathcal{A}};\pi_t^{\smile}\sqcap(\rho_{s_1}\times\cdots\times\rho_{s_n});f^{\mathcal{B}};\rho_t);\pi_t & \pi_t \text{ univalent}\\
= & f^{\mathcal{P}};\pi_t &
\end{array}$$

The direct product properties all follow from the component-wise definitions of the operations involved. $\square$



Interestingly, even if every product in $\mathbf{D}$ contains at least one surjective projection (as for example in *Rel*), this need not be the case in $\Sigma$-$\mathsf{Alg}_{\mathbf{D}}$.

For a simple example, consider the following two $\mathsf{sigTwoSets}$-algebras $\mathcal{A}$ and $\mathcal{B}$:

- $\mathcal{P}^{\mathcal{A}} = \{0\}$, $\mathcal{Q}^{\mathcal{A}} = \emptyset$
- $\mathcal{P}^{\mathcal{B}} = \emptyset$, $\mathcal{Q}^{\mathcal{B}} = \{1\}$

Their direct product has empty sets for both carriers, but neither of the projections is surjective.

Chapter 4

# Dedekind Categories of Graph Structures

While in general, subalgebra lattices need not even be modular, we have seen in Sect. 2.4 that subalgebra lattices of graph structures are completely distributive complete lattices. In the last chapter, we have established that relational homomorphisms between $\Sigma$-algebras form locally co-complete allegories — this implies that homsets are lattices, but there is not very much known about the structure of these lattices.

In this section we show how for unary signatures, we obtain complete distributivity in the homsets, and also the corresponding distributivity over composition. Therefore, for every unary signature $\Sigma$, the unary $\Sigma$-algebras together with relational $\Sigma$-algebra homomorphisms form a *Dedekind category*.

Dedekind categories are, informally speaking, heterogeneous relation algebras without complement. So they still impose quite a lot of structure, and it is not obvious how Dedekind categories might be a useful framework for the study of rewriting.

Hitherto, Dedekind categories have been studied mostly in the context of fuzzy relations, for which they form a useful abstraction, see [Fur98, KFM99]. There, representation theorems are given which state that Dedekind categories where certain point axioms involving "crispness" concepts hold are equivalent to matrix algebras over lattices of "scalars". Crispness and this kind of point axioms turn out to be rather natural in the context of fuzzy relations.

In the Dedekind categories of graph structures that we present in this chapter, frequently only universal relations are crisp, and points are rare, so their additional structure is relatively uncharted.

We start with introducing the necessary background on Dedekind categories. We then show in Sect. 4.2 that even when considering general abstract $\Sigma$-algebras over a Dedekind category, we do not obtain a Dedekind category again. For this, the restriction to unary signatures is sufficient (and, in general, necessary), as we shall see in Sect. 4.3.

We then show how pseudo-complements and semi-complements can be calculated in Dedekind categories of graph structures.

In addition to the constructions from Sect. 3.3, these Dedekind categories also permit direct sum constructions, as we shall see in Sect. 4.5.

Finally, we explore how the concepts of discreteness and solidity behave in concert with relational homomorphisms.

## 4.1 Preliminaries: Distributive Allegories and Dedekind Categories

To the structure presented so far, we now add the possibility of finding joins and a zero together with distributivity of composition over joins.

**Definition 4.1.1** A *distributive allegory* is a tuple
$\mathbf{C} = (Obj_{\mathbf{C}}, Mor_{\mathbf{C}}, \_ : \_ \leftrightarrow \_, \mathbb{I}, ;, \breve{~}, \sqcap, \sqcup, \perp\!\!\!\perp)$ where the following hold:





- The tuple $(Obj_{\mathbf{C}}, Mor_{\mathbf{C}}, \_ : \_ \leftrightarrow \_, \mathbb{I}, ;, \breve{~}, \sqcap)$ is an allegory, the so-called *underlying allegory* of $\mathbf{C}$.
- Every homset $\mathsf{Mor}_{\mathcal{C}}[\mathcal{A},\mathcal{B}]$ carries the structure of a distributive lattice with $\sqcup_{\mathcal{A},\mathcal{B}}$ for *join*, and zero element $\perp\!\!\!\perp_{\mathcal{A},\mathcal{B}}$.
- For all objects $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ and all morphisms $Q : \mathcal{A} \leftrightarrow \mathcal{B}$, the *zero law* holds:

$$Q;\perp\!\!\!\perp_{\mathcal{B},\mathcal{C}} = \perp\!\!\!\perp_{\mathcal{A},\mathcal{C}} \ .$$

- For all $Q : \mathcal{A} \leftrightarrow \mathcal{B}$ and $R, R' : \mathcal{B} \leftrightarrow \mathcal{C}$, *join-distributivity* holds:

$$Q;(R \sqcup R') = Q;R \sqcup Q;R' \ . \qquad \square$$

Interaction between arbitrary joins and composition is covered by the concept of *local completeness*, for which we just reformulate the definition of [FS90, 2.22]:

**Definition 4.1.2** A distributive allegory is *locally complete* if every homset is a completely upwards-distributive complete lattice, and if composition distributes over arbitrary unions, that is, given $Q : \mathcal{A} \leftrightarrow \mathcal{B}$ and a subset $\mathcal{R}$ of $\mathsf{Mor}[\mathcal{B},\mathcal{C}]$, one has

$$Q;(\bigsqcup \mathcal{R}) = \bigsqcup\{R : \mathcal{R} \bullet Q;R\} \ . \qquad \square$$

For our concept of local co-completeness (Def. 3.1.6) to continue to make sense in distributive allegories, we have to cover the interaction between joins and arbitrary meets, too:

**Definition 4.1.3** A distributive allegory $\mathbf{D}$ is called *locally co-complete* iff its underlying allegory is locally co-complete, and every homset lattice is completely downwards-distributive.

**Definition 4.1.4** Let a category $\mathbf{C}$ be given.

- $\mathbf{C}$ *has right residuals* if for every two morphisms $P : \mathcal{A} \to \mathcal{B}$ and $R : \mathcal{A} \to \mathcal{C}$, their *right residual* $P\backslash R : \mathcal{B} \to \mathcal{C}$ exists, where the right residual is defined by

$$P;X \sqsubseteq R \qquad \Leftrightarrow \qquad X \sqsubseteq P\backslash R \qquad \text{for all } X : \mathcal{B} \to \mathcal{C}.$$

- $\mathbf{C}$ *has left residuals* if for every two morphisms $Q : \mathcal{B} \to \mathcal{C}$ and $R : \mathcal{A} \to \mathcal{C}$, their *left residual* $R/Q : \mathcal{A} \to \mathcal{B}$ exists, where the left residual is defined by

$$Y;Q \sqsubseteq R \qquad \Leftrightarrow \qquad Y \sqsubseteq R/Q \qquad \text{for all } Y : \mathcal{A} \to \mathcal{B}. \qquad \square$$

In allegories, existence of left residuals follows via conversion from existence of right residuals, and vice versa.

It is easy to see that locally complete distributive allegories have residuals.

Olivier and Serrato introduced Dedekind categories as categories with residuals, where homsets are complete lattices, and where there is a conversion operation for which a modal rule holds [OS80]. Today, it is well-known that Dedekind categories are exactly locally complete distributive allegories with explicitly listed residuals.

For us, the precise distinction between primitive and derived operations is irrelevant, so we define:

**Definition 4.1.5** A *Dedekind category* is a locally complete distributive allegory. □

In the sequel we use the name “Dedekind category” mostly because it is well-introduced, and considerably shorter than “locally complete distributive allegory”.

Residuals allow us to directly define useful operations that provide access to that part of a relation that is univalent respectively injective from the point of view of the domain:

**Definition 4.1.6** In a Dedekind category, for a relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ we define

- its *univalent part* $\mathsf{upa}\, R : \mathcal{A} \leftrightarrow \mathcal{B}$ as $\mathsf{upa}\, R := R \sqcap R^{\smile} \backslash \mathbb{I}$,
- its *domain of injectivity* $\mathsf{injdom}\, R : \mathsf{PId}\, \mathcal{A}$ as $\mathsf{injdom}\, R := \mathbb{I} \sqcap (R;R^{\smile}) \backslash \mathbb{I}$, and
- its *injective part* $\mathsf{ipa}\, R : \mathcal{A} \leftrightarrow \mathcal{B}$ as $\mathsf{ipa}\, R := (\mathsf{injdom}\, R);R$. □

Note that with our definition here, the injective part is *not* just the converse of the univalent part of the converse! The domain of injectivity $\mathsf{injdom}\, R$ may in general be smaller than the range of $\mathsf{upa}\,(R^{\smile})$. In Lemma A.3.1 we show that the univalent part deserves its name; see also [SS93, 4.2.8].

**Lemma 4.1.7** [←192]

- If $R$ is total, then $\mathsf{injdom}\, R = (R;R^{\smile}) \backslash \mathbb{I}$.
- $\mathsf{ipa}\, R$ is injective.

**Proof**:

- If $R$ is total, then $\mathbb{I} \sqsubseteq R;R^{\smile}$, and since the right residual is antitonic in its left argument, we have $(R;R^{\smile}) \backslash \mathbb{I} \sqsubseteq \mathbb{I} \backslash \mathbb{I} = \mathbb{I}$.

- $$\begin{array}{rcll} \mathsf{ipa}\, R;(\mathsf{ipa}\, R)^{\smile} & = & (\mathsf{injdom}\, R);R;R^{\smile};\mathsf{injdom}\, R & \\ & = & (\mathsf{injdom}\, R);R;R^{\smile};(\mathbb{I} \sqcap (R;R^{\smile}) \backslash \mathbb{I}) & \\ & \sqsubseteq & (\mathsf{injdom}\, R);R;R^{\smile};((R;R^{\smile}) \backslash \mathbb{I}) & \\ & \sqsubseteq & \mathsf{injdom}\, R;\mathbb{I} & \text{residual property} \\ & \sqsubseteq & \mathbb{I} & \end{array}$$ □

Local completeness implies that in Dedekind categories, homsets are pseudo-complemented lattices; however, we reserve shall rarely need relative pseudo-complements in the lattices of complete homsets, and frequently need relative pseudo-complements in the sublattices of partial identities. Therefore, we introduce a new symbol $\Rightarrow$ to denote relative pseudo-complement of arbitrary relations, and reserve the conventional notations for use in the sublattices of partial identities. The notation $p \rightarrow q$ of Def. 2.6.1 will only be used for partial identities $p$ and $q$ on the same object, say $\mathcal{A}$, and denotes the partial identity on $\mathcal{A}$ which is the relative pseudo-complements of $p$ wrt. $q$ in the lattice of partial identities on $\mathcal{A}$. In the same way, we restrict the notation $r^{\neg}$ for pseudo-complements to partial identities. These operations can be defined via $\Rightarrow$; for every two partial identities $p, q : \mathsf{PId}\, \mathcal{A}$ we define:

$$p \rightarrow q := \mathbb{I} \sqcap (p \Rightarrow q) \ , \qquad p^{\neg} := p \rightarrow \perp\!\!\!\perp = \mathbb{I} \sqcap (p \Rightarrow \perp\!\!\!\perp) \ .$$

Since we even more frequently need semi-complements, we introduce a stronger variant of Dedekind categories:

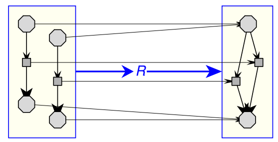


**Definition 4.1.8** A *strict Dedekind category* is a locally complete and locally co-complete distributive allegory. □

In a strict Dedekind category, homsets are therefore also semi-complemented lattices, and we use the notation $R \setminus S$ of Def. 2.7.1 also for relative semi-complements of arbitrary relations. If $p, q : \mathsf{PId}\, \mathcal{A}$ are partial identities, then $p \setminus q$ is a partial identity, too, so we need not make a notational distinction here. However, we reserve the semi-complement notation $p^{\sim}$ to be used only in lattices of partial identities, so we have $p^{\sim} := \mathbb{I} \setminus p$.

We now introduce a concept that is closely related with semi-complements, and which is going to be extremely important in the context of graph transformation.

When injectivity of a relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is required only on a part of $\mathcal{A}$, the idiom “$q;R$ is injective” for some partial identity $q$ is often not sufficiently expressive. For example, the following relational graph homomorphism is injective on all edges, but there is no partial identity that includes only edges, but no vertices.




This morphism is, however, *almost-injective besides* the partial identity containing all vertices of $\mathcal{A}$ (the discrete base of $\mathcal{A}$), in the sense of the following definition:

**Definition 4.1.9** Given two partial identities $u, v : \mathsf{PId}\, \mathcal{A}$, a relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is called *almost-injective on $u$ besides $v$* iff

$$u;R;R^{\smile} \sqsubseteq u \sqcup v;R;R^{\smile} \ .$$

Furthermore, $R$ is called

- *almost-injective on $u$* if $R$ is almost-injective on $u$ besides $u^{\sim}$,
- *almost-injective besides $v$* if $R$ is almost-injective on $v^{\sim}$ besides $v$, and
- *almost-injective* if $R$ is almost-injective on $\mathsf{dom}\, R$ besides $(\mathsf{dom}\, R)^{\sim}$. □

Note that this definition has a shape that does not allow to relate any kind of almost-injectivity of $R$ on $u$ with injectivity of $u;R$.

Obviously, if $R$ is almost-injective on $u$ besides $v$ and $u \sqsubseteq \mathsf{dom}\, R$, then we have the equality

$$u;R;R^{\smile} = u \sqcup u;v;R;R^{\smile} \ .$$

Furthermore, we have:

**Lemma 4.1.10** For two partial identities $u, v : \mathsf{PId}\, \mathcal{A}$ with $u \sqcup v \sqsupseteq \mathbb{I}$, a relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is almost-injective on $u$ besides $v$, iff

$$u;R;R^{\smile} \sqsubseteq u \sqcup v;R;R^{\smile};v \ ,$$

and therefore also iff

$$R;R^{\smile} = u;\mathsf{dom}\, R \sqcup v;R;R^{\smile};v \ .$$

**Proof**: "⇐" is obvious, so we only need to show "⇒":

$$\begin{array}{rll} u;R;R^{\smile} & \sqsubseteq \; u \sqcup v;R;R^{\smile} & R \text{ almost-injective on } u \text{ besides } v \\ & = \; u \sqcup v;R;R^{\smile};u \sqcup v;R;R^{\smile};v & \mathbb{I} = u \sqcup v \\ & \sqsubseteq \; u \sqcup u \sqcup v;R;R^{\smile};v \sqcup v;R;R^{\smile};v & R \text{ almost-injective on } u \text{ besides } v \\ & = \; u \sqcup v;R;R^{\smile};v & \end{array}$$

□

## 4.2 Joins in General Σ-Algebra Allegories

The existence of arbitrary meets, as seen above in Proposition 3.2.5 for $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ over locally co-complete $\mathbf{D}$, implies that, in analogy to Theorem 2.3.6, there also is a relational morphism closure for Σ-compatible families of relations between to Σ-algebras. This closure is, in the same way as the subalgebra closure, defined as a meet. If we now also have joins available in $\mathbf{D}$, then we may also obtain it via joins, similar to the joins in Theorem 2.3.6.

**Theorem 4.2.1** [←93] Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a Dedekind category $\mathbf{D}$ be given. If $\mathcal{A}$ and $\mathcal{B}$ are abstract Σ-algebras over $\mathbf{D}$, and $R := (R_t)_{t:\mathcal{S}}$ is a family of relations such that $R_t : t^{\mathcal{A}} \leftrightarrow t^{\mathcal{B}}$, then we define the *relational morphism closure* of $R$ as the least relational Σ-algebra homomorphism $C$ from $\mathcal{A}$ to $\mathcal{B}$ such that $R_t \sqsubseteq C_t$ for every sort $t : \mathcal{S}$, or equivalently as the following meet:

$$\mathsf{RMC}_{\Sigma,\mathbf{D}}(R) := \bigsqcap\{Q : \mathcal{A} \leftrightarrow \mathcal{B} \mid (\forall s : \mathcal{S} \bullet R_s \sqsubseteq Q_s)\} \ .$$

Then $\mathsf{RMC}_{\Sigma,\mathbf{D}}(R)$ is the least Σ-compatible family $C := (C_t)_{t:\mathcal{S}}$ of relations such that for every sort $t : \mathcal{S}$ the following holds:

$$C_t = R_t \sqcup \bigsqcup\{\vec{s} : \mathcal{S}^*; f : \mathcal{F} \mid f : \vec{s} \to t \bullet (f^{\mathcal{A}})^{\smile};C_{\vec{s}};f^{\mathcal{B}}\}$$

**Proof**: First we have to show that the two variants of the definition are equivalent, that is, that the set $\mathcal{Q} := \{Q : \mathcal{A} \leftrightarrow \mathcal{B} \mid (\forall s : \mathcal{S} \bullet R_s \sqsubseteq Q_s)\}$ does indeed have a least element. For this, it is sufficient to show $\bigsqcap \mathcal{Q} \in \mathcal{Q}$. Since with Proposition 3.2.5, meets may be taken component-wise, we have the following for every sort $t : \mathcal{S}$:

$$\begin{array}{rcl} (\bigsqcap \mathcal{Q})_t & = & \bigsqcap\{Q : \mathcal{Q} \bullet Q_t\} = \bigsqcap\{Q : \mathcal{A} \leftrightarrow \mathcal{B} \mid (\forall s : \mathcal{S} \bullet R_s \sqsubseteq Q_s) \bullet Q_t\} \\ & \sqsupseteq & \bigsqcap\{T : t^{\mathcal{A}} \leftrightarrow t^{\mathcal{B}} \mid R_t \sqsubseteq T\} = R_t \end{array}$$

Therefore, $\bigsqcap \mathcal{Q} \in \mathcal{Q}$, and we may turn to showing the join formulation:

$$\begin{array}{lll} & R \sqsubseteq C & \\ \Leftrightarrow & (\forall t : \mathcal{S} \bullet R_t \sqsubseteq C_t) \quad \wedge \quad C \text{ is } \Sigma\text{-morphism} & \\ \Leftrightarrow & (\forall t : \mathcal{S} \bullet R_t \sqsubseteq C_t) \quad \wedge \quad (\forall \vec{s} : \mathcal{S}^*; t : \mathcal{S}; f : \mathcal{F} \mid f : \vec{s} \to t \bullet C_{\vec{s}};f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}};C_t) & \\ \Leftrightarrow & \forall t : \mathcal{S} \bullet R_t \sqsubseteq C_t \quad \wedge \quad \forall \vec{s} : \mathcal{S}^*; f : \mathcal{F} \mid f : \vec{s} \to t \bullet C_{\vec{s}};f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}};C_t & \\ \Leftrightarrow & \forall t : \mathcal{S} \bullet R_t \sqsubseteq C_t \quad \wedge \quad \forall \vec{s} : \mathcal{S}^*; f : \mathcal{F} \mid f : \vec{s} \to t \bullet (f^{\mathcal{A}})^{\smile};C_{\vec{s}};f^{\mathcal{B}} \sqsubseteq C_t & \text{A.1.2.iii)} \\ \Leftrightarrow & \forall t : \mathcal{S} \bullet R_t \sqcup \bigsqcup\{\vec{s} : \mathcal{S}^*; f : \mathcal{F} \mid f : \vec{s} \to t \bullet (f^{\mathcal{A}})^{\smile};C_{\vec{s}};f^{\mathcal{B}}\} \sqsubseteq C_t & \end{array}$$

□



The importance of this join formulation lies in the fact that it can easily be used to formulate algorithms for calculating relational morphism closures. If Σ is acyclic, then a topological ordering of the sorts produces a program with a primitively-recursive top-level structure on top of operations in $\mathbf{D}$. This is the case for graphs since $\mathsf{sigGraph}$ is acyclic, and we obtain the following ($\mathcal{E}$ and $\mathcal{V}$ are the sorts for edges and vertices, and $s$ and $t$ are the function symbols for source and target of edges):

$$\begin{array}{rcl} (\mathsf{RMC}_{\Sigma,\mathbf{D}}(R))_{\mathcal{E}} & = & R_{\mathcal{E}} \\ (\mathsf{RMC}_{\Sigma,\mathbf{D}}(R))_{\mathcal{V}} & = & R_{\mathcal{V}} \sqcup (s^{\mathcal{A}})^{\smile};R_{\mathcal{E}};s^{\mathcal{B}} \sqcup (t^{\mathcal{A}})^{\smile};R_{\mathcal{E}};t^{\mathcal{B}} \end{array}$$

If the signature has cycles, then the necessary fixpoint iterations can be tamed by performing them separately inside every strongly connected component, and globally proceeding along a topological ordering of the strongly connected components.

Since the existence of arbitrary meets in $\Sigma\text{-}\mathsf{Alg}_{\mathbf{D}}$ over locally co-complete $\mathbf{D}$ implies the existence of arbitrary joins, the above theorem also provides the way to calculate these joins:

**Corollary 4.2.2** [←93] Let a signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, a Dedekind category $\mathbf{D}$, and two abstract Σ-algebras $\mathcal{A}$ and $\mathcal{B}$ over $\mathbf{D}$ be given.

If $\mathcal{R}$ is a set of relational Σ-algebra homomorphisms from $\mathcal{A}$ to $\mathcal{B}$, then

$$\bigsqcup \mathcal{R} = \mathsf{RMC}_{\Sigma,\mathbf{D}}((\bigsqcup\{R : \mathcal{R} \bullet R_t\})_{t:\mathcal{S}}) \ .$$

In particular, $\quad \mathbb{L}_{\mathcal{A},\mathcal{B}} = \mathsf{RMC}_{\Sigma,\mathbf{D}}((\mathbb{L}_{t^{\mathcal{A}},t^{\mathcal{B}}})_{t:\mathcal{S}}) \quad$ and $\quad R \sqcup S = \mathsf{RMC}_{\Sigma,\mathbf{D}}((R_t \sqcup S_t)_{t:\mathcal{S}})$ . □

Since the simplest join is the empty join, let us first consider least elements in the homsets of Σ-algebra allegories. Here, the zero law of distributive allegories does not hold in the presence of constants; for an example of this effect consider the algebra $\mathcal{A}$ over the signature $\mathsf{sigC1}$ where the carrier has two elements: $\mathcal{N}^{\mathcal{A}} = \{0, 1\}$, and the constant is one of them: $c^{\mathcal{A}} = 1$. Then it is easy to see that $\mathbb{L}_{\mathcal{N}^{\mathcal{A}},\mathcal{N}^{\mathcal{A}}} = \{(1, 1)\}$, and with $R := \{(0, 1), (1, 1)\}$ we have $R;\mathbb{L} = R \neq \mathbb{L}$.

Non-empty joins are by their definition least upper bounds wrt. inclusion, and therefore naturally obey the lattice laws. However, in the presence of binary operators, lattice distributivity need not hold: since the subalgebra lattice is isomorphic to the lattice of partial identities, a counterexample has been seen on page 40. Join-distributivity of composition need not hold, either, as the following (computer-generated) example demonstrates.

**Example 4.2.3** Consider the $\mathsf{sigB1}$-algebras $\mathcal{A}$ and $\mathcal{B}$ with $\mathcal{N}^{\mathcal{A}} = \mathcal{N}^{\mathcal{B}} = \{1, 2\}$, and the following interpretations of the binary function symbol $f$:

$$f^{\mathcal{A}}(x, y) = 2 \qquad\qquad f^{\mathcal{B}}(x, y) = \begin{cases} 1 & \text{if } x = 2 \text{ and } y = 1 \\ 2 & \text{otherwise} \end{cases}$$

Let us now consider the following relational homomorphisms:

$$\begin{array}{rl} R : \mathcal{B} \leftrightarrow \mathcal{B} & R \;=\; \{(2, 1), (2, 2)\} \\ S : \mathcal{B} \leftrightarrow \mathcal{B} & S \;=\; \{(1, 1), (2, 2)\} \;=\; \mathbb{I} \\ Q : \mathcal{A} \leftrightarrow \mathcal{B} & Q \;=\; \{(1, 1), (2, 1), (2, 2)\} \end{array}$$

Then $Q;(R \sqcup S) = \mathbb{T} \not\sqsubseteq Q = Q;R \sqcup Q;S$. □

## 4.3 Relational Homomorphisms Between Graph Structures

Graph structures are $\Sigma$-algebras over unary signatures, and unary signatures have been introduced in Def. 2.2.1 as special cases of general signatures. Since from now on, we shall be concerned only with unary signatures, we may as well give a specialised, simpler definition:

**Definition 4.3.1** A *unary signature* is a graph $(\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ consisting of

- a set $\mathcal{S}$ of *sorts*,
- a set $\mathcal{F}$ of *function symbols*,
- a mapping $\mathsf{src} : \mathcal{F} \to \mathcal{S}$ associating with every function symbol its *source sort*, and
- a mapping $\mathsf{trg} : \mathcal{F} \to \mathcal{S}$ associating with every function symbol its *target sort*. □

Of course, we continue to write "$f : s \to t$" for a function symbol $f \in \mathcal{F}$ with $\mathsf{src}(f) = s$ and $\mathsf{trg}(f) = t$. We now also consider all unary signatures introduced before as unary signatures in the sense of this definition.

The definition of a signature as a graph (obviously, $\mathcal{S}$ is the interpretation of $\mathcal{V}$, and $\mathcal{F}$ that of $\mathcal{E}$) allows us to use the fact that contained in every Dedekind category $\mathbf{D}$ there is the category $\mathsf{Map}\,\mathbf{D}$ having the same objects, but only all mappings as arrows, and underlying this category is a graph that ignores composition and identities. This gives rise to a slightly more abstract definition of abstract unary $\Sigma$-algebras:

**Definition 4.3.2** Given an allegory $\mathbf{D}$ and a unary signature $\Sigma$, an *abstract unary $\Sigma$-algebra* is a graph homomorphism from $\Sigma$ to $\mathsf{Map}\,\mathbf{D}$. □

More intuitively, this means that for every abstract unary $\Sigma$-algebra $\mathcal{A}$, there are (we continue to use the notation of Def. 3.2.1)

- for every sort $s \in \mathcal{S}$, an object $s^{\mathcal{A}} \in Obj_{\mathbf{D}}$, and
- for every function symbol $f \in \mathcal{F}$ with $f : s \to t$ a mapping $f^{\mathcal{A}} : s^{\mathcal{A}} \to t^{\mathcal{A}}$ in $\mathsf{Map}\,\mathbf{D}$.

This corresponds to the view of Def. 3.2.1, except that we do not need any products anymore.

We also restate the definition of relational homomorphisms, since the homomorphism condition now looks much simpler:

**Definition 4.3.3** [←22] Given a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and two abstract unary $\Sigma$-algebras $\mathcal{A}$ and $\mathcal{B}$ over $\mathbf{D}$, a *relational $\Sigma$-algebra homomorphism from $\mathcal{A}$ to $\mathcal{B}$* is an $\mathcal{S}$-indexed family of relations $\Phi = (\Phi_s)_{s \in \mathcal{S}}$ such that

- $\Phi_s : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{B}}$, and
- for all $f \in \mathcal{F}$ with $f : s \to t$ we have $\Phi_s ; f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}} ; \Phi_t$. □

Accordingly, also the proofs of the category and allegory properties carry less syntactical ballast; we restate them for the benefit of readers who skipped the last chapter:



**Proposition 4.3.4** Given an allegory $\mathbf{D}$ and a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, relational $\Sigma$-algebra homomorphisms form a category, where composition and identities are defined component-wise.

**Proof**: First we show that $\mathbb{I}_{\mathcal{A}}$ is a relational $\Sigma$-algebra homomorphism:

- $(\mathbb{I}_{\mathcal{A}})_s = \mathbb{I}_{s^{\mathcal{A}}} : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{A}}$, and
- for all $f \in \mathcal{F}$ with $f : s \to t$ we have $(\mathbb{I}_{\mathcal{A}})_s ; f^{\mathcal{A}} = \mathbb{I}_{s^{\mathcal{A}}} ; f^{\mathcal{A}} = f^{\mathcal{A}} ; \mathbb{I}_{t^{\mathcal{A}}} = f^{\mathcal{A}} ; (\mathbb{I}_{\mathcal{A}})_t$.

Now we show well-definedness of composition: Let $\Psi : \mathcal{A} \to \mathcal{B}$ and $\Psi : \mathcal{B} \to \mathcal{B}$ be relational $\Sigma$-algebra homomorphisms, and $\Xi := \Phi ; \Psi$, then:

- $\Xi_s = \Phi_s ; \Psi_s : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{C}}$, and
- for all $f \in \mathcal{F}$ with $f : s \to t$ we have

$$\Xi_s ; f^{\mathcal{C}} = \Phi_s ; \Psi_s ; f^{\mathcal{C}} \sqsubseteq \Phi_s ; f^{\mathcal{B}} ; \Psi_t \sqsubseteq f^{\mathcal{A}} ; \Phi_t ; \Psi_t = f^{\mathcal{A}} ; \Xi_t \ .$$

Associativity of composition and the identity laws follow via the component-wise definitions. □

**Definition 4.3.5** [←91] Given a distributive allegory $\mathbf{D}$ and a unary signature $\Sigma$, we define the following operations on relational $\Sigma$-algebra homomorphisms:

- If $\Phi : \mathcal{A} \to \mathcal{B}$, then the *converse* of $\Phi = (\Phi_s)_{s \in \mathcal{S}}$ is $\Phi^{\smile} := (\Phi_s^{\smile})_{s \in \mathcal{S}}$.
- If $\Phi, \Psi : \mathcal{A} \to \mathcal{B}$, then their *meet* and *join* are defined component-wise:

$$(\Phi \sqcap \Psi)_s := \Phi_s \sqcap \Psi_s \qquad \text{and} \qquad (\Phi \sqcup \Psi)_s := \Phi_s \sqcup \Psi_s$$

- $\perp\!\!\!\perp_{\mathcal{A},\mathcal{B}} := (\perp\!\!\!\perp_{s^{\mathcal{A}},s^{\mathcal{B}}})_{s \in \mathcal{S}}$. □

**Proposition 4.3.6** The operations of Def. 4.3.5 are well-defined.

**Proof**: For converse, we need the fact that $f^{\mathcal{A}}$ and $f^{\mathcal{B}}$ are mappings, for being able to apply Lemma A.1.2.iii):

$$(\Phi^{\smile})_s ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{B}} ; (\Phi^{\smile})_t \Leftrightarrow \Phi_s^{\smile} ; f^{\mathcal{A}} \sqsubseteq f^{\mathcal{B}} ; \Phi_t^{\smile} \Leftrightarrow f^{\mathcal{B}\smile} ; \Phi_s^{\smile} \sqsubseteq \Phi_t^{\smile} ; f^{\mathcal{A}\smile} \Leftrightarrow \Phi_s ; f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}} ; \Phi_t$$

For meet, we need univalence of $f^{\mathcal{A}}$.

$$\begin{array}{lclclclcl}
(\Phi \sqcap \Psi)_s ; f^{\mathcal{B}} & = & (\Phi_s \sqcap \Psi_s) ; f^{\mathcal{B}} & \sqsubseteq & \Phi_s ; f^{\mathcal{B}} \sqcap \Psi_s ; f^{\mathcal{B}} & \sqsubseteq & f^{\mathcal{A}} ; \Phi_t \sqcap f^{\mathcal{A}} ; \Psi_t & = & f^{\mathcal{A}} ; (\Phi \sqcap \Psi)_t \\
(\Phi \sqcup \Psi)_s ; f^{\mathcal{B}} & = & (\Phi_s \sqcup \Psi_s) ; f^{\mathcal{B}} & = & \Phi_s ; f^{\mathcal{B}} \sqcup \Psi_s ; f^{\mathcal{B}} & \sqsubseteq & f^{\mathcal{A}} ; \Phi_t \sqcup f^{\mathcal{A}} ; \Psi_t & = & f^{\mathcal{A}} ; (\Phi \sqcup \Psi)_t
\end{array}$$

The zero law is essential for the empty relation: $\perp\!\!\!\perp ; f^{\mathcal{B}} = \perp\!\!\!\perp = f^{\mathcal{A}} ; \perp\!\!\!\perp$. □

Given the closedness of relational $\Sigma$-algebra homomorphisms under converse, meet, and join, properties of relations for these operations are inherited by relational $\Sigma$-algebra homomorphisms because of the component-wise definition.

Therefore, we immediately see that abstract unary $\Sigma$-algebras with relational homomorphisms form a distributive allegory, where the inclusion ordering between standard homomorphisms is again defined component-wise (and we shall write $\sqsubseteq$ for this ordering, too).

**Theorem 4.3.7** Abstract unary $\Sigma$-algebras over a distributive allegory $\mathbf{D}$ together with relational unary $\Sigma$-algebra homomorphisms form a distributive allegory. $\square$

It is easy to see that well-definedness of joins and meets continues to hold when generalised to the infinite variants:

**Definition 4.3.8** [←93] Given a strict Dedekind category $\mathbf{D}$, a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, and a set $\mathcal{R}$ of relational $\Sigma$-algebra homomorphisms such that for every $\Phi \in \mathcal{R}$ we have $\Phi : \mathcal{A} \to \mathcal{B}$, then the meet and join over the set $\mathcal{R}$ are defined component-wise:

$$(\bigsqcap \mathcal{R})_s = \bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_s\} \qquad \text{and} \qquad (\bigsqcup \mathcal{R})_s = \bigsqcup\{\Phi : \mathcal{R} \bullet \Phi_s\}$$

**Proof of well-definedness**:

$$\begin{array}{rcll}
(\bigsqcap \mathcal{R})_s ; f^{\mathcal{B}} & = & (\bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_s\}) ; f^{\mathcal{B}} & \\
& \sqsubseteq & \bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_s ; f^{\mathcal{B}}\} & \text{local co-completeness} \\
& \sqsubseteq & \bigsqcap\{\Phi : \mathcal{R} \bullet f^{\mathcal{A}} ; \Phi_t\} & \\
& = & f^{\mathcal{A}} ; (\bigsqcap\{\Phi : \mathcal{R} \bullet \Phi_t\}) & f^{\mathcal{A}} \text{ univalent, Lemma A.1.5} \\
& = & f^{\mathcal{A}} ; (\bigsqcap \mathcal{R})_t & \\
(\bigsqcup \mathcal{R})_s ; f^{\mathcal{B}} & = & (\bigsqcup\{\Phi : \mathcal{R} \bullet \Phi_s\}) ; f^{\mathcal{B}} & \\
& \sqsubseteq & \bigsqcup\{\Phi : \mathcal{R} \bullet \Phi_s ; f^{\mathcal{B}}\} & \text{local completeness} \\
& \sqsubseteq & \bigsqcup\{\Phi : \mathcal{R} \bullet f^{\mathcal{A}} ; \Phi_t\} & \\
& = & f^{\mathcal{A}} ; (\bigsqcup\{\Phi : \mathcal{R} \bullet \Phi_t\}) & \text{local completeness} \\
& = & f^{\mathcal{A}} ; (\bigsqcup \mathcal{R})_t &
\end{array}$$
$\square$

Therefore, we also inherit local (co-)completeness from the component relations.

Local completeness in turn implies existence of residuals. However, we do not inherit a component-wise definition of residuals, since such a component-wise residual is in general not a legal homomorphism. (Where the component-wise residual *is* a homomorphism, it is of course also the residual of homomorphisms.)

Relational unary $\Sigma$-algebra homomorphisms, even between abstract unary $\Sigma$-algebras over a relation algebra[1], are also *not* (in general) closed under component-wise complementation. Even given the smallest and largest homomorphisms $\mathbb{⊥}$ and $\mathbb{T}$, the ordering $\sqsubseteq$ is not complementary.

Therefore we have:

**Theorem 4.3.9 (Dedekind category of graph structures)** For every unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$, abstract unary $\Sigma$-algebras over a strict Dedekind category $\mathbf{D}$ together with relational unary $\Sigma$-algebra homomorphisms form a strict Dedekind category, a *Dedekind category of graph structures*, denoted $\Sigma\text{-}\mathsf{GS}_{\mathbf{D}}$. $\square$

[1] A *relation algebra* is a Dedekind category where every homset is a Boolean lattice.



## 4.4 Pseudo- and Semi-Complements in $\Sigma$-$\mathsf{GS}_{\mathbf{D}}$

Remember that subalgebra closure (Theorem 2.3.6) is natural for arbitrary signatures, while the subalgebra kernel (Theorem 2.4.6) required unary signatures. In the same way, the dual of Theorem 4.2.1 only works well for unary signatures:

**Theorem 4.4.1** Let a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a Dedekind category $\mathbf{D}$ be given. If $\mathcal{A}$ and $\mathcal{B}$ are abstract $\Sigma$-algebras over $\mathbf{D}$, and $R := (R_t)_{t:\mathcal{S}}$ is a family of relations such that $R_t : t^{\mathcal{A}} \leftrightarrow t^{\mathcal{B}}$, then we define the *relational morphism kernel* of $R$ as the greatest relational $\Sigma$-algebra homomorphism $K$ from $\mathcal{A}$ to $\mathcal{B}$ such that $K_s \sqsubseteq R_s$ for every sort $s : \mathcal{S}$, or equivalently as the following join:

$$\mathsf{RMK}_{\Sigma,\mathbf{D}}(R) := \bigsqcup\{Q : \mathcal{A} \leftrightarrow \mathcal{B} \mid (\forall t : \mathcal{S} \bullet Q_t \sqsubseteq R_t)\} \ .$$

Then $\mathsf{RMK}_{\Sigma,\mathbf{D}}(R)$ is the greatest $\Sigma$-compatible family $K := (K_s)_{s:\mathcal{S}}$ of relations such that for every sort $s : \mathcal{S}$ the following holds:

$$K_s = R_s \sqcap \bigsqcap\{t : \mathcal{S}; f : \mathcal{F} \mid f : s \to t \bullet f^{\mathcal{A}} ; K_t ; (f^{\mathcal{B}})^{\smile}\}$$

**Proof**: First we have to show that the two variants of the definition are equivalent, that is, that the set $\mathcal{Q} := \{Q : \mathcal{A} \leftrightarrow \mathcal{B} \mid (\forall t : \mathcal{S} \bullet Q_t \sqsubseteq R_t)\}$ does indeed have a greatest element. For this, it is sufficient to show $\bigsqcup \mathcal{Q} \in \mathcal{Q}$. Since with Def. 4.3.8, joins may be taken component-wise, we have the following for every sort $s : \mathcal{S}$:

$$\begin{array}{rcl}
(\bigsqcup \mathcal{Q})_s & = & \bigsqcup\{Q : \mathcal{Q} \bullet Q_s\} = \bigsqcup\{Q : \mathcal{A} \leftrightarrow \mathcal{B} \mid (\forall t : \mathcal{S} \bullet Q_t \sqsubseteq R_t) \bullet Q_s\} \\
& \sqsubseteq & \bigsqcup\{S : s^{\mathcal{A}} \leftrightarrow s^{\mathcal{B}} \mid S \sqsubseteq R_s\} \\
& = & R_s
\end{array}$$

Therefore, $\bigsqcup \mathcal{Q} \in \mathcal{Q}$, and we may turn to showing the join formulation:

$$\begin{array}{cll}
& K \sqsubseteq R & \\
\Leftrightarrow & (\forall s : \mathcal{S} \bullet K_s \sqsubseteq R_s) \quad \wedge \quad K \text{ is } \Sigma\text{-morphism} & \\
\Leftrightarrow & (\forall s : \mathcal{S} \bullet K_s \sqsubseteq R_s) \quad \wedge \quad (\forall s, t : \mathcal{S}; f : \mathcal{F} \mid f : s \to t \bullet K_s ; f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}} ; K_t) & \\
\Leftrightarrow & \forall s : \mathcal{S} \bullet K_s \sqsubseteq R_s \quad \wedge \quad \forall t : \mathcal{S}; f : \mathcal{F} \mid f : s \to t \bullet K_s ; f^{\mathcal{B}} \sqsubseteq f^{\mathcal{A}} ; K_t & \\
\Leftrightarrow & \forall s : \mathcal{S} \bullet K_s \sqsubseteq R_s \quad \wedge \quad \forall t : \mathcal{S}; f : \mathcal{F} \mid f : s \to t \bullet K_s \sqsubseteq f^{\mathcal{A}} ; K_t ; (f^{\mathcal{B}})^{\smile} & \text{A.1.2.iii)} \\
\Leftrightarrow & \forall s : \mathcal{S} \bullet K_s \sqsubseteq R_s \sqcap \bigsqcap\{t : \mathcal{S}; f : \mathcal{F} \mid f : s \to t \bullet f^{\mathcal{A}} ; K_t ; (f^{\mathcal{B}})^{\smile}\} &
\end{array}$$
$\square$

For non-unary signatures, what fails is the equivalence of the two definition variants: as we have seen in Corollary 4.2.2, for non-unary signatures, joins are not defined component-wise, and $\mathcal{Q}$ will, in general, not have a greatest element. In such a case, the join in the definition of $\mathsf{RMK}_{\Sigma,\mathbf{D}}(R)$ is not contained in $R$, while the meet formulation always is contained in $R$ — it then calculates the intersection of all maximal elements of $\mathcal{Q}$.

For unary signatures, the relational morphism kernel can, by this theorem, be calculated essentially in the same way as the closure, and reduces to a simple shape for acyclic signatures. For graphs, we have:

$$\begin{array}{rcl}
(\mathsf{RMK}_{\Sigma,\mathbf{D}}(R))_{\mathcal{V}} & = & R_{\mathcal{V}} \\
(\mathsf{RMK}_{\Sigma,\mathbf{D}}(R))_{\mathcal{E}} & = & R_{\mathcal{E}} \sqcap s^{\mathcal{A}} ; R_{\mathcal{V}} ; (s^{\mathcal{B}})^{\smile} \sqcap t^{\mathcal{A}} ; R_{\mathcal{V}} ; (t^{\mathcal{B}})^{\smile}
\end{array}$$

Because of the dual definitions of pseudo-complements and semi-complements we then obviously have the following dual expressions:

**Corollary 4.4.2** [←96] Let a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a strict Dedekind category $\mathbf{D}$ be given. If $\mathcal{A}$ and $\mathcal{B}$ are abstract $\Sigma$-algebras over $\mathbf{D}$, and $Q, R$ are relational $\Sigma$-algebra homomorphisms from $\mathcal{A}$ to $\mathcal{B}$, then

$$Q \Rrightarrow R = \mathsf{RMK}_{\Sigma,\mathbf{D}}((Q_s \Rrightarrow R_s)_{s:\mathcal{S}}) \qquad \text{and} \qquad Q \setminus R = \mathsf{RMC}_{\Sigma,\mathbf{D}}((Q_s \setminus R_s)_{s:\mathcal{S}}) \ . \quad \square$$

## 4.5 Constructions in $\Sigma$-$\mathsf{GS}_{\mathbf{D}}$

The constructions of Sect. 3.3 can of course all be used for graph structures, too. Direct definitions of the constructions and proofs of their correctness would be much simpler since with unary signatures, there is no need to handle argument tuples. However, there is no additional insight to be gained, so we do not re-state the definitions and proofs for subobjects, quotients, and direct products.

Let us only mention that in a direct product graph structure $\mathcal{P}$ for $\mathcal{A}$ and $\mathcal{B}$, the interpretation of a function symbol $f : s \to t$ is now a simple direct product of the interpretations in $\mathcal{A}$ and $\mathcal{B}$:

$$f^{\mathcal{P}} = f^{\mathcal{A}} \times f^{\mathcal{B}} = \pi_s ; f^{\mathcal{A}} ; \pi_t^{\smile} \sqcap \rho_s ; f^{\mathcal{B}} ; \rho_t^{\smile}$$

One construction has been conspicuously missing in Sect. 3.3, namely that of direct sums: Since the characterisation of direct sums involves joins, it could not be treated in the raw allegory setting.

**Definition 4.5.1** In a distributive allegory, a *direct sum* for two objects $\mathcal{A}$ and $\mathcal{B}$ is a triple $(\mathcal{S}, \iota, \kappa)$ consisting of an object $\mathcal{S}$ and two *injections*, i.e., relations $\iota : \mathcal{A} \leftrightarrow \mathcal{S}$ and $\kappa : \mathcal{B} \leftrightarrow \mathcal{S}$ for which the following conditions hold:

$$\iota ; \iota^{\smile} = \mathbb{I}_{\mathcal{A}} \ , \qquad \kappa ; \kappa^{\smile} = \mathbb{I}_{\mathcal{B}} \ , \qquad \iota ; \kappa^{\smile} = \bot\!\!\!\bot_{\mathcal{A},\mathcal{B}} \ , \qquad \iota^{\smile} ; \iota \sqcup \kappa^{\smile} ; \kappa = \mathbb{I}_{\mathcal{S}} \ . \qquad \square$$

For graph structures, direct sums can be constructed component-wise[2]:

**Theorem 4.5.2** Let a unary signature $\Sigma = (\mathcal{S}, \mathcal{F}, \mathsf{src}, \mathsf{trg})$ and a strict Dedekind category $\mathbf{D}$ be given. Let $\mathcal{A}$ and $\mathcal{B}$ be two objects of $\Sigma$-$\mathsf{GS}_{\mathbf{D}}$. If for every sort $s \in \mathcal{S}$ there is a direct sum $(\mathcal{C}_s, \iota_{\mathcal{C}_s}, \kappa_{\mathcal{C}_s})$ for $\mathcal{A}_s$ and $\mathcal{B}_s$ in $\mathbf{D}$, then there is also a direct sum $(\mathcal{C}, \iota_{\mathcal{C}}, \kappa_{\mathcal{C}})$ for $\mathcal{A}$ and $\mathcal{B}$ in $\Sigma$-$\mathsf{GS}_{\mathbf{D}}$.

Therefore, if $\mathbf{D}$ has direct sums, then $\Sigma$-$\mathsf{GS}_{\mathbf{D}}$ has direct sums, too.

**Proof**: For every sort $s \in \mathcal{S}$, choose a direct sum $(\mathcal{C}_s, \iota_{\mathcal{C}_s}, \kappa_{\mathcal{C}_s})$. Let $\mathcal{C}$ be defined by these $\mathcal{C}_s$ as carriers and by defining for every function symbol $f : s \to t$ the mapping

$$f^{\mathcal{C}} := \iota_{\mathcal{C}_s}^{\smile} ; f^{\mathcal{A}} ; \iota_{\mathcal{C}_t} \sqcup \kappa_{\mathcal{C}_s}^{\smile} ; f^{\mathcal{B}} ; \kappa_{\mathcal{C}_t}$$

All these $f^{\mathcal{C}}$ are by definition total and univalent.

Furthermore, $\iota_{\mathcal{C}} := (\iota_{\mathcal{C}_s})_{s \in \mathcal{S}}$ and $\kappa_{\mathcal{C}} := (\kappa_{\mathcal{C}_s})_{s \in \mathcal{S}}$ are by definition total, univalent and injective, and they are also relational homomorphisms, as we show only for $\iota_{\mathcal{C}}$:

$$\iota_{\mathcal{C}_s} ; f^{\mathcal{C}} = \iota_{\mathcal{C}_s} ; \iota_{\mathcal{C}_s}^{\smile} ; f^{\mathcal{A}} ; \iota_{\mathcal{C}_t} \sqcup \iota_{\mathcal{C}_s} ; \kappa_{\mathcal{C}_s}^{\smile} ; f^{\mathcal{B}} ; \kappa_{\mathcal{C}_t} = f^{\mathcal{A}} ; \iota_{\mathcal{C}_t}$$

The direct sum properties all follow from the component-wise definitions of the operations involved. $\square$

[2] It is well-known that in the category $\mathsf{Map}(\Sigma\text{-}\mathsf{Alg}_{\mathcal{S}et})$ of conventional $\Sigma$-homomorphisms, there are categorical sums, but they cannot be constructed component-wise: the carriers have to be closed recursively under the images of more-than-unary operators in their applications to elements of different components.



## 4.6 Discrete Relations

In Sect. 2.9, we introduced concepts like *discreteness* and *solid parts* for elements of completely distributive complete lattices. Now we have such lattices as homsets of strict Dedekind categories, so we could also use all these concepts for arbitrary relations. However, it seems more natural to keep the application of these concepts restricted to the domain of substructures they had been designed for. Therefore, we shall use all the concepts of Sect. 2.9 only inside lattices of partial identities in Dedekind categories, so we may talk about *discrete partial identities*, or about one partial identity *being a solid part of* another. Since we have a one-to-one correspondence between subalgebras of a (concrete) algebra $\mathcal{A}$ and partial identities on $\mathcal{A}$ in the corresponding concrete Dedekind category, the explanations of Sect. 2.9 carry directly over to this setting.

Furthermore, predicates that have been declared for whole lattices in Sect. 2.9, are now carried over to objects, too. For example, $\mathcal{A}$ is *connected* iff $(\mathsf{PId}\,\mathcal{A}, \sqsubseteq)$ is connected as a lattice (see Def. 2.9.6).

Starting from this basis, we now introduce further concepts in this context for morphisms of Dedekind categories, i.e., for (abstract) relations.

**Definition 4.6.1** A relation $Q : \mathcal{A} \leftrightarrow \mathcal{B}$ is called *range-discrete* iff $\mathsf{ran}\,Q$ is discrete, and *domain-discrete* iff $\mathsf{dom}\,Q$ is discrete. $\square$

**Definition 4.6.2** A Dedekind category is called *discreteness-preserving* iff for all relations $Q : \mathcal{A} \leftrightarrow \mathcal{B}$ and $S : \mathcal{B} \leftrightarrow \mathcal{C}$ we have

$$\begin{aligned} Q \text{ range-discrete} &\;\Rightarrow\; Q;S \text{ range-discrete} \\ S \text{ domain-discrete} &\;\Rightarrow\; Q;S \text{ domain-discrete} \end{aligned} \qquad \square$$

In a discreteness-preserving Dedekind category we have equivalence of domain- and range-discreteness:

$$\begin{aligned} Q \text{ range-discrete} &\;\Rightarrow\; Q;Q^{\smile} \text{ range-discrete} \\ &\;\Leftrightarrow\; \mathsf{ran}\,(Q;Q^{\smile}) \text{ discrete} \\ &\;\Leftrightarrow\; \mathsf{ran}\,Q^{\smile} \text{ discrete} \\ &\;\Leftrightarrow\; \mathsf{dom}\,Q \text{ discrete} \\ &\;\Leftrightarrow\; Q \text{ domain-discrete} \\ &\;\Rightarrow\; Q \text{ range-discrete} \qquad \text{analogously} \end{aligned}$$

Therefore, we define a common name:

**Definition 4.6.3** In a discreteness-preserving Dedekind category, a relation $Q$ is called *discrete* if it is range-discrete. $\square$

Note that this notion of discreteness concepts for (heterogeneous) relational (graph) homomorphisms is not related with discreteness concepts for homogeneous relations that directly represent graphs, as for example in [SS93, Def. 6.5.4].

From the definition it is obvious that for a discrete relation $Q$ in a discreteness-preserving Dedekind category also its converse $Q^{\smile}$ and arbitrary compositions $P{:}Q$ and $Q{:}S$ are discrete, and also arbitrary meets, since every relation $R \sqsubseteq Q$ is discrete, too. Even joins preserve discreteness:

**Lemma 4.6.4** If for a subset $\mathcal{Q}$ of a homset $\mathsf{Mor}[\mathcal{A},\mathcal{B},]$ every element of $\mathcal{Q}$ is range-discrete, than $\bigsqcup \mathcal{Q}$ is range-discrete, too.

**Proof**: Range-discreteness of $\bigsqcup \mathcal{Q}$ follows from Lemma 2.9.3 since

$$\mathsf{ran}\,(\bigsqcup \mathcal{Q}) = \bigsqcup\{Q : \mathcal{Q} \bullet \mathsf{ran}\,Q\} \ . \qquad \square$$

This demonstrates that the join over all discrete partial identities on an object is discrete, too, and therefore allows us to define:

**Definition 4.6.5** For an object $\mathcal{A}$ in a Dedekind category, we denoted with $\mathbb{D}_{\mathcal{A}} : \mathsf{PId}\,\mathcal{A}$ the *discrete base* of $\mathcal{A}$, defined as the maximal discrete partial identity on $\mathcal{A}$:

$$\mathbb{D}_{\mathcal{A}} := \bigsqcup\{q : \mathsf{PId}\,\mathcal{A} \mid q \text{ discrete}\} \qquad \square$$

When we consider $\Sigma$-$\mathsf{GS}_{\mathbf{R}}$ over a relation algebra $\mathbf{R}$, then $\Phi_t \setminus \Xi_t = \Phi_t \sqcap \overline{\Xi_t}$, and also

$$\Phi_t \setminus (\Phi_t \setminus \Xi_t) = \Phi_t \sqcap \overline{\Phi_t \sqcap \overline{\Xi_t}} = \Phi_t \sqcap (\overline{\Phi_t} \sqcup \Xi_t) = \Phi_t \sqcap \Xi_t \ .$$

However, from the relational morphism closure formulation for the semi-complement (Corollary 4.4.2) we easily see that

$$\Phi \setminus (\Phi \setminus \Xi) = \Phi \sqcap \Xi$$

will hold in general only under the following restrictions:

(i) $\Sigma$ is acyclic

(ii) $\Phi_s = \mathbb{⊥}$ if $s$ is not a sink sort, i.e., $\Phi_s = \mathbb{⊥}$ whenever there is some $f \in \mathcal{F}$ such that $f : s \to t$ for some $t \in \mathcal{S}$.

Item (ii) characterises discrete morphisms in $\Sigma$-$\mathsf{GS}_{\mathbf{R}}$ if $\Sigma$ is acyclic, and the following result is then obvious:

**Theorem 4.6.6** If $\mathbf{R}$ is a relation algebra and $\Sigma$ is an acyclic graph, then $\Sigma$-$\mathsf{GS}_{\mathbf{R}}$ is discreteness-preserving. $\square$

Nevertheless, $\Sigma$-$\mathsf{GS}_{\mathbf{R}}$ will only be border-discrete if in addition there are no sorts in $\Sigma$ that are source and target of different function symbols:



**Theorem 4.6.7** If $\mathbf{R}$ is a relation algebra and $\Sigma$ is a bipartite graph, then $\Sigma$-$\mathsf{GS}_{\mathbf{R}}$ is border-discrete. $\square$

One way to obtain such strong results for more complicated signatures $\Sigma$ is to restrict the unary algebras under consideration: Together with $\Sigma$-$\mathsf{GS}_{\mathbf{D}}$, also every full subcategory is a Dedekind category.

For example, simple sink sorts may be replaced by strongly connected components for which all internal cycles are always interpreted as identities.

**Definition 4.6.8** For a partial identity $q : \mathsf{PId}\,\mathcal{A}$, we let $\mathsf{sol}\,q : \mathsf{PId}\,\mathcal{A}$ denote its *solid part*, defined as

$$\mathsf{sol}\,q := \bigsqcup\{p : \mathsf{PId}\,\mathcal{A} \mid p \sqsubseteq q, \text{ and } p \text{ solid}\} \ . \qquad \square$$

**Lemma 4.6.9** If $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is an arbitrary relation in a border-discrete Dedekind category, then

$$\mathsf{ran}\,(R{:}(\mathsf{ran}\,R)^{\sim}) \sqsubseteq \mathsf{sol}\,((\mathsf{ran}\,R)^{\sim})$$

**Proof**: This follows immediately from Lemma 2.9.13. $\square$

**Lemma 4.6.10** Let a discrete relation $Q : \mathcal{A} \leftrightarrow \mathcal{B}$ and a partial identity $r : \mathsf{PId}\,\mathcal{B}$ be given. If there is a solid partial identity $q : \mathsf{PId}\,\mathcal{B}$ such that $q \sqsubseteq r$ and $\mathsf{ran}\,Q \sqcap r \sqsubseteq q$, then $q \setminus \mathsf{ran}\,Q = q$ and $r \setminus \mathsf{ran}\,Q = r$, implying in particular $(\mathsf{ran}\,Q)^{\sim} = \mathbb{I}$.

**Proof**: Since discreteness of $Q$ means that $\mathsf{ran}\,Q$ is discrete, this follows immediately from Lemma 2.9.15. $\square$

**Lemma 4.6.11** [←97] If $q : \mathsf{PId}\,\mathcal{A}$ is solid and $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is univalent and $q \sqsubseteq \mathsf{dom}\,R$, then $\mathsf{ran}\,(q{:}R)$ is solid, too.

**Proof**: Assume $u : \mathsf{PId}\,\mathcal{B}$ is discrete, and $u \sqsubseteq \mathsf{ran}\,(q{:}R)$. Then:

$$\begin{array}{rcll}
(\mathsf{ran}\,(q{:}R)) \setminus u \sqsubseteq Y & \Leftrightarrow & \mathsf{ran}\,(q{:}R) \sqsubseteq Y \sqcup u & \\
& \Rightarrow & q \sqsubseteq \mathsf{ran}\,(Y{:}R^{\smile}) \sqcup \mathsf{ran}\,(u{:}R^{\smile}) & q \sqsubseteq \mathsf{dom}\,R \\
& \Leftrightarrow & q \sqsubseteq \mathsf{ran}\,(Y{:}R^{\smile}) \sqcup \mathsf{dom}\,(R{:}u) & \\
& \Leftrightarrow & q \setminus \mathsf{dom}\,(R{:}u) \sqsubseteq \mathsf{ran}\,(Y{:}R^{\smile}) & \\
& \Leftrightarrow & q \sqsubseteq \mathsf{ran}\,(Y{:}R^{\smile}) & q \text{ solid, } R{:}u \text{ discrete} \\
& \Rightarrow & \mathsf{ran}\,(q{:}R) \sqsubseteq Y & R \text{ univalent}
\end{array}$$

This shows $\mathsf{ran}\,(q{:}R) \sqsubseteq (\mathsf{ran}\,(q{:}R)) \setminus u$, and we have equality since the opposite inclusion is trivial. Therefore, $\mathsf{ran}\,(q{:}R)$ is solid, too. $\square$

**Definition 4.6.12** A relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$ is called *solid* iff for every solid partial identity $q : \mathsf{PId}\,\mathcal{A}$, its image $\mathsf{ran}\,(q{:}R)$ is solid, too. $\square$

Lemma 4.6.11 shows that mappings are solid. In general, however, we need not demand totality; it is sufficient if $\mathsf{sol}\,\mathbb{I} \sqsubseteq \mathsf{dom}\,R$, and it should also be possible to find a more general condition than univalence.

## Chapter 5

# Categoric Rewriting in a Relational Setting

Most of the categoric approach to graph transformation relies on the category-theoretic notion of pushouts, and our aim is, roughly, to provide a variant of the double-pushout approach that accommodates "graph variables" and relational matching, and where replication of the images of variables will be possible.

Replication is notoriously impossible with pushouts, but is inherent in the dual concept of pullbacks. Accordingly, there is a lesser-known variant of the categoric approach that uses pullbacks for graph transformation, put forward by Bauderon and Jacquet [Jac99, Bau97, BJ96, BJ01].

However, that approach never really gained popularity, probably mostly because its rules as such appear to be quite unintuitive and usually have to be regarded as "encodings" of the rules of other approaches. With that encoding attitude, however, the pullback approach is able to cover most popular approaches to graph rewriting, including the double-pushout approach.

We now study pushouts and pullbacks of mappings and the respective complements, and also pushouts of partial functions, in the relational setting. Pullbacks are covered by the concept of *tabulation*, a name introduced by Freyd and Scedrov [FS90, 2.14]. Relational characterisations of pushouts and pushout complements have been achieved by Kawahara [Kaw90], using the setting of relations in a countably complete topos. This setting can, at least for our purposes here, safely be considered as a special case of the Dedekind category setting.

For pullback complements and pushouts of partial functions we are not aware of any previous relational characterisation.

Although pushouts are more wide-spread in graph rewriting and maybe easier to understand, it turns out that pullbacks can already be handled appropriately on the allegory level while pushouts require transitive closures, and thus local completeness. Therefore, we start with pullbacks in an allegory setting, and thereafter treat pushouts in a Dedekind category setting.

Throughout this chapter, we employ the simple but useful category-theoretic concepts of *span* and *cospan* to achieve more concise formulations.

A *span* is an ordered pair $(f, g)$ of morphisms $f : \mathcal{A} \to \mathcal{B}$ and $g : \mathcal{A} \to \mathcal{C}$ with the same source. Such a span is often written $\mathcal{B} \xleftarrow{f} \mathcal{A} \xrightarrow{g} \mathcal{C}$.

Analogously, a *cospan* is an ordered pair $(h, k)$ of morphisms $h : \mathcal{B} \to \mathcal{D}$ and $k : \mathcal{C} \to \mathcal{D}$ with the same target, written $\mathcal{B} \xrightarrow{h} \mathcal{D} \xleftarrow{k} \mathcal{C}$.

Since allegories and Dedekind categories are categories, too, we may use these notions for arbitrary relations with common source or target, respectively. Usually, the components of a (co)span are taken from the allegory or Dedekind category of the current discussion.

In this chapter we investigate category-theoretic concepts in their application to the category of mappings contained in an allegory. Therefore, we frequently have to explicitly





state if a span is intended to be in the mapping category $\mathsf{Map}\,\mathbf{D}$ instead of in the allegory or Dedekind category $\mathbf{D}$ of the current discussion.

## 5.1 Pullbacks

The category-theoretic notion of pullback is one special case of what is called a "limit" in category theory:

**Definition 5.1.1** [←16] In a category $\mathbf{C}$, a *pullback* for a cospan $\mathcal{B} \xrightarrow{h} \mathcal{D} \xleftarrow{k} \mathcal{C}$ is a span $\mathcal{B} \xleftarrow{f} \mathcal{A} \xrightarrow{g} \mathcal{C}$, such that $f;h = g;k$, and that for every span $\mathcal{B} \xleftarrow{f'} \mathcal{A} \xrightarrow{g'} \mathcal{C}$ with $f';h = g';k$ there is a unique morphism $u : \mathcal{A}' \to \mathcal{A}$ such that $f' = u;f$ and $g' = u;g$. □

$$\begin{tikzcd}
\mathcal{D} & \mathcal{B} \arrow[l, "h"'] & \\
\mathcal{C} \arrow[u, "k"] & \mathcal{A} \arrow[l, "g"'] \arrow[u, "f"'] & \\
 & & \mathcal{A}' \arrow[uul, "f'"'] \arrow[ull, "g'"] \arrow[ul, dotted, "u"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{D} & \mathcal{B} \arrow[l, "R"'] \arrow[dl, "W"'] & \\
\mathcal{C} \arrow[u, "S"] & \mathcal{A} \arrow[l, "Q"'] \arrow[u, "P"'] & \\
 & & \mathcal{A}' \arrow[uul, "P'"'] \arrow[ull, "Q'"] \arrow[ul, dotted, "U"']
\end{tikzcd}$$

By a standard argument over this universal characterisation, pullbacks are unique up to isomorphism: Assume two pullbacks $\mathcal{B} \xleftarrow{f} \mathcal{A} \xrightarrow{g} \mathcal{C}$ and $\mathcal{B} \xleftarrow{f'} \mathcal{A}' \xrightarrow{g'} \mathcal{C}$ for $\mathcal{B} \xrightarrow{h} \mathcal{D} \xleftarrow{k} \mathcal{C}$, then there are $u : \mathcal{A}' \to \mathcal{A}$ with $f' = u;f$ and $g' = u;g$ and $v : \mathcal{A} \to \mathcal{A}'$ with $f = v;f'$ and $g = v;g'$. Then $f' = u;f = u;v;f'$ and $g' = u;g = u;v;g'$. Therefore, $u;v$ factorises the second pullback via itself, but since $\mathbb{I}_{\mathcal{A}'}$ also does this, unique factorisation implies $u;v = \mathbb{I}_{\mathcal{A}'}$. In the same way one obtains $v;u = \mathbb{I}_{\mathcal{A}}$, so $u$ and $v$ are isomorphisms.

An extreme case of pullbacks is where $\mathcal{D}$ is a terminal object in $\mathbf{C}$ — the pullback of two (unique) morphisms to a terminal object is a categorical product.

Correspondingly, a relational characterisation of pullbacks should generalise the relational characterisation of direct products. This generalisation is called *tabulation* by Freyd and Scedrov [FS90, 2.14]; we here give a definition that is equivalent, but has a shape that is a direct generalisation of the definition of direct products we use (Def. 3.1.10, page 70).

**Definition 5.1.2** [←70, 146] In an allegory $\mathbf{D}$, let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

The span $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ in the *allegory* $\mathbf{D}$ (i.e., $P$ and $Q$ are not yet specified as mappings) is a *direct tabulation* for $W$ iff the following equations hold:

$$P^{\smile};Q = W \qquad P^{\smile};P = \mathsf{dom}\,W \qquad Q^{\smile};Q = \mathsf{ran}\,W \qquad P;P^{\smile} \sqcap Q;Q^{\smile} = \mathbb{I} \qquad \square$$

Even though $P$ and $Q$ were not specified as mappings to start with, their totality follows from the last equation, and their univalence from the second and third.

Freyd and Scedrov specify $P$ and $Q$ as mappings and omit our second and third equations — they follow via univalence and totality from the first.

However, on the one hand these equations establish the correspondence with our direct product characterisation, which is just a direct tabulation for a universal relation, and on the other hand we feel that they are also useful in calculations using tabulations, as we shall see below.

First we show a generalised factorisation property that unifies part of the proofs of the following two theorems.

**Lemma 5.1.3** [←100, 101] In an allegory $\mathbf{D}$, let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

If the span $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ is a direct tabulation of $W$, and if the span $\mathcal{B} \xleftarrow{P'} \mathcal{A}' \xrightarrow{Q'} \mathcal{C}$ of mappings satisfies the following conditions:

$$Q' \sqsubseteq P'{;}W \qquad \text{and} \qquad P' \sqsubseteq Q'{;}W^{\smile} ,$$

then $U := P'{;}P^{\smile} \sqcap Q'{;}Q^{\smile}$ is a mapping from $\mathcal{A}'$ to $\mathcal{A}$ such that $P' = U{;}P$ and $R' = U{;}R$.

**Proof**: We only show factorisation of $P'$; factorisation of $Q'$ follows in the same way.

$$\begin{array}{rcll} U{;}P & = & (P'{;}P^{\smile} \sqcap Q'{;}Q^{\smile}){;}P & \\ & = & P' \sqcap Q'{;}Q^{\smile}{;}P & P \text{ univalent, Lemma A.1.2.ii)} \\ & = & P' \sqcap Q'{;}W^{\smile} & \text{direct tabulation} \\ & = & P' & P' \sqsubseteq Q'{;}W^{\smile} \end{array}$$

Via factorisation, totality of $U$ follows from totality of $P'$ (or $Q'$):

$$\mathsf{dom}\, U \sqsupseteq \mathsf{dom}\,(U{;}P) = \mathsf{dom}\, P' = \mathbb{I}$$

With univalence of $P'$ and $Q'$ we obtain univalence of $U$:

$$\begin{array}{rcl} U^{\smile}{;}U & = & (P{;}P'^{\smile} \sqcap Q{;}Q'^{\smile}){;}(P'{;}P^{\smile} \sqcap Q'{;}Q^{\smile}) \\ & \sqsubseteq & P{;}P'^{\smile}{;}P'{;}P^{\smile} \sqcap Q{;}Q'^{\smile}{;}Q'{;}Q^{\smile} \;\sqsubseteq\; P{;}P^{\smile} \sqcap Q{;}Q^{\smile} \;=\; \mathbb{I} \end{array}$$ □

The characterisation of direct tabulations is monomorphic; the following theorem corresponds to [FS90, 2.144]:

**Theorem 5.1.4** In an allegory $\mathbf{D}$, let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

If the spans $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ and $\mathcal{B} \xleftarrow{P'} \mathcal{A}' \xrightarrow{Q'} \mathcal{C}$ (as spans of relations in $\mathbf{D}$) are both direct tabulations of $W$, then $U := P'{;}P^{\smile} \sqcap Q'{;}Q^{\smile}$ is a bijective mapping from $\mathcal{A}'$ to $\mathcal{A}$ such that $P' = U{;}P$ and $R' = U{;}R$.

**Proof**: Since $\mathcal{B} \xleftarrow{P'} \mathcal{A}' \xrightarrow{Q'} \mathcal{C}$ is a direct tabulation, $P'$ and $Q'$ are mappings and we have $P'^{\smile}{;}Q' = W$; so we easily obtain the preconditions of Lemma 5.1.3:

$$Q' \sqsubseteq P'{;}P'^{\smile}{;}Q' = P'{;}W \qquad \text{and} \qquad P' \sqsubseteq Q'{;}Q'^{\smile}{;}P' = Q'{;}W^{\smile} .$$

Lemma 5.1.3 then shows that $U$ is a mapping that factorises $P'$ and $Q'$; since the same argument is valid for $U^{\smile}$, too, we also know that $U$ is bijective. □



The connection with pullbacks arises since given a cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$, every tabulation for the relation $R{;}S^{\smile}$ is a pullback in the category of mappings:

**Theorem 5.1.5** Let $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ be objects in an allegory $\mathbf{D}$, and let $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ be a cospan in $\mathsf{Map}\,\mathbf{D}$, i.e., $R : \mathcal{B} \leftrightarrow \mathcal{D}$ and $S : \mathcal{C} \leftrightarrow \mathcal{D}$ are mappings. A direct tabulation $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ for $W := R{;}S^{\smile}$ is then a pullback in $\mathsf{Map}\,\mathbf{D}$ for $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$.

**Proof**: The mapping properties of $P$ and $Q$ follow directly from the tabulation properties. Totality of $P$ and $Q$ and univalence of $R$ and $S$ together with the first tabulation property show commutativity:

$$\begin{array}{rcccccccl} P{;}R & \sqsubseteq & Q{;}Q^{\smile}{;}P{;}R & = & Q{;}W^{\smile}{;}R & = & Q{;}S{;}R^{\smile}{;}R & \sqsubseteq & Q{;}S \\ & \sqsubseteq & P{;}P^{\smile}{;}Q{;}S & = & P{;}W{;}S & = & P{;}R{;}S^{\smile}{;}S & \sqsubseteq & P{;}R \end{array}$$

Assume that there is a span $\mathcal{B} \xleftarrow{P'} \mathcal{A}' \xrightarrow{Q'} \mathcal{C}$ in $\mathsf{Map}\,\mathbf{D}$ with $P'{;}R = Q'{;}S$. Then define:

$$U := P'{;}P^{\smile} \sqcap Q'{;}Q^{\smile}$$

With totality of $R$ and $S$ and commutativity for $\mathcal{B} \xleftarrow{P'} \mathcal{A}' \xrightarrow{Q'} \mathcal{C}$ we easily obtain the preconditions of Lemma 5.1.3:

$$\begin{array}{rcl} Q' & \sqsubseteq & Q'{;}S{;}S^{\smile} = P'{;}R{;}S^{\smile} = P'{;}W \\ P' & \sqsubseteq & P'{;}R{;}R^{\smile} = Q'{;}S{;}R^{\smile} = Q'{;}W^{\smile} \end{array}$$

Lemma 5.1.3 then shows that $U$ is a mapping that factorises $P'$ and $Q'$. So we only need to show that $U$ is uniquely determined. Assume a mapping $U' : \mathcal{A}' \to \mathcal{A}$ with $U'{;}P = P'$ and $U'{;}Q = Q'$. Univalence of $U'$ and the fourth tabulation property then yield equality with $U$:

$$U' = U'{;}(P{;}P^{\smile} \sqcap Q{;}Q^{\smile}) = U'{;}P{;}P^{\smile} \sqcap U'{;}Q{;}Q^{\smile} = P'{;}P^{\smile} \sqcap Q'{;}Q^{\smile} = U .$$ □

We now show how the usual pullback construction of the category *Set* of sets and concrete mappings may be reformulated in the allegory-theoretic setting, relying only on one direct product and one subobject.

**Definition 5.1.6** Let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be a relation in an allegory $\mathbf{D}$. If there exists a sharp direct product $(\mathcal{P}, \pi, \rho)$ for $\mathcal{B}$ and $\mathcal{C}$, and if a subobject injection $\lambda : \mathcal{A} \leftrightarrow \mathcal{P}$ exists for the partial identity

$$\mathsf{dom}\,(\pi{;}W \sqcap \rho)$$

(this means that $\lambda$ is an injective mapping and $\mathsf{ran}\,\lambda = \mathsf{dom}\,(\pi{;}W \sqcap \rho)$), we call the span $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ with

$$P := \lambda{;}\pi \qquad \text{and} \qquad Q := \lambda{;}\rho$$

a *constructed tabulation* for $W$. □

It is straightforward to show that constructed tabulations are well-defined:

**Theorem 5.1.7** Every constructed tabulation $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ for $W$ is a direct tabulation for $W$.

**Proof**: $P$ and $Q$ are mappings by definition. With Lemma A.2.2.v) we have:

$$\mathsf{ran}\,\lambda = \mathsf{dom}\,(\pi;W \sqcap \rho) = \mathbb{I} \sqcap \pi;W;\rho^{\smile} = \mathbb{I} \sqcap \rho;W^{\smile};\pi^{\smile} \;.$$

With univalence of $\pi$ and $\rho$ and Lemma A.1.2.ii) this implies:

$$\begin{aligned} (\mathsf{ran}\,\lambda);\rho &= (\mathbb{I} \sqcap \pi;W;\rho^{\smile});\rho = \rho \sqcap \pi;W \\ (\mathsf{ran}\,\lambda);\pi &= (\mathbb{I} \sqcap \rho;W^{\smile};\pi^{\smile});\pi = \pi \sqcap \rho;W^{\smile} \end{aligned}$$

For $P^{\smile};Q = W$ we may use the fact that there is a universal relation between $\mathcal{B}$ and $\mathcal{C}$ because of existence of the direct product:

$$W = W \sqcap \mathbb{T}_{\mathcal{B},\mathcal{C}} = W \sqcap \pi^{\smile};\rho = \pi^{\smile};(\pi;W \sqcap \rho) = \pi^{\smile};(\mathsf{ran}\,\lambda);\rho = \pi^{\smile};\lambda^{\smile};\lambda;\rho = P^{\smile};Q$$

The tabulation equation follows from the tabulation equation in the definition of the direct product:

$$P;P^{\smile} \sqcap Q;Q^{\smile} = \lambda;\pi;\pi^{\smile};\lambda^{\smile} \sqcap \lambda;\rho;\rho^{\smile};\lambda^{\smile} = \lambda;(\pi;\pi^{\smile} \sqcap \rho;\rho^{\smile});\lambda^{\smile} = \lambda;\lambda^{\smile} = \mathbb{I}$$

Finally we have, using $(\mathsf{ran}\,\lambda);\pi = \pi \sqcap \rho;W^{\smile}$ from above:

$$\begin{aligned} P^{\smile};P &= \pi^{\smile};\lambda^{\smile}\lambda;\pi \\ &= (\pi^{\smile} \sqcap W;\rho^{\smile});\lambda^{\smile}\lambda;(\pi \sqcap \rho;W^{\smile}) \\ &= (\pi^{\smile} \sqcap W;\rho^{\smile});\mathsf{ran}\,\lambda;(\pi \sqcap \rho;W^{\smile}) \\ &= (\pi^{\smile} \sqcap W;\rho^{\smile});(\pi \sqcap \rho;W^{\smile}) && \mathsf{dom}\,(\pi \sqcap \rho;W^{\smile}) = \mathsf{ran}\,\lambda \\ &= \mathbb{I} \sqcap W;W^{\smile} && \text{sharp product} \\ &= \mathsf{dom}\,W \end{aligned}$$

In the same way, we also obtain $Q^{\smile};Q = \mathsf{ran}\,W$. □

Note that this construction requires the existence of the sharp direct product $\mathcal{B} \times \mathcal{C}$ as an intermediate object. In non-standard allegories, it is perfectly possible that a tabulation exists although this product may not exist.

In addition, in extreme cases there may be some cospan in $\mathsf{Map}\,\mathbf{D}$ which has a pullback (in $\mathsf{Map}\,\mathbf{D}$) which is not a tabulation, in the same way as there may be categorical products in $\mathsf{Map}\,\mathbf{D}$ which are not direct products. However, these foundational issues will not be of importance in our context.

## 5.2 Transitive and Difunctional Closures

In Dedekind categories, local completeness implies that the transitive closure exists for every morphism $R : \mathcal{A} \leftrightarrow \mathcal{A}$, since it can be defined in the following way:

$$R^{+} := \bigsqcap\{X : \mathcal{A} \leftrightarrow \mathcal{A} \mid R \sqsubseteq X \wedge X;X \sqsubseteq X\} = \bigsqcup\{i : \mathbb{N} \mid i \geq 1 \bullet R^{i}\}$$

The reflexive and transitive closure is then, as usual, $R^{*} := \mathbb{I} \sqcup R^{+}$. We now define three derived operations which are useful abbreviations, especially when their operands are larger terms:



**Definition 5.2.1** For $R : \mathcal{A} \leftrightarrow \mathcal{B}$ we define $R^{\rhd} : \mathcal{A} \leftrightarrow \mathcal{A}$ and $R^{\lhd} : \mathcal{B} \leftrightarrow \mathcal{B}$, and the *difunctional closure* $R^{\boxast} : \mathcal{A} \leftrightarrow \mathcal{B}$ as

$$R^{\rhd} := (R;R^{\smile})^{*} \;, \qquad R^{\lhd} := (R^{\smile};R)^{*} \;, \qquad R^{\boxast} := R^{\rhd};R = R;R^{\lhd} \;.$$ □

A relation $R$ is called *difunctional* iff $R;R^{\smile};R \sqsubseteq R$ (this inclusion is equivalent to equality). It is easy to see that the difunctional closure deserves its name: If $R \sqsubseteq Q$ and $Q$ is difunctional, then $Q;Q^{\smile};Q \sqsubseteq Q$ implies $R;R^{\smile};R \sqsubseteq Q$, and further $R;R^{\smile};R;R^{\smile};R \sqsubseteq Q$, and so on, and therefore

$$R^{\boxast} = (R;R^{\smile})^{*};R \sqsubseteq Q \;.$$

On the other hand, we have

$$R^{\boxast};(R^{\boxast})^{\smile};R^{\boxast} = (R;R^{\smile})^{*};R;R^{\smile};(R;R^{\smile})^{*};(R;R^{\smile})^{*};R = (R;R^{\smile})^{+};R \sqsubseteq R^{\boxast} \;,$$

so $R^{\boxast}$ is itself difunctional, and therefore $R^{\boxast}$ is the least difunctional relation containing $R$. (See [SS93, 4.4] for more about difunctionality.)

By definition, we furthermore have $R^{\lhd} = (R^{\smile})^{\rhd}$.

## 5.3 Pushouts

Much of the categoric approach to graph rewriting is based on the concept of pushout, which is just the dual of the pullback:

**Definition 5.3.1** [↔4] In a category, a *pushout* for a span $\mathcal{B} \xleftarrow{f} \mathcal{A} \xrightarrow{g} \mathcal{C}$ is a cospan $\mathcal{B} \xrightarrow{h} \mathcal{D} \xleftarrow{k} \mathcal{C}$, such that $f;h = g;k$, and that for every cospan $\mathcal{B} \xrightarrow{h'} \mathcal{D}' \xleftarrow{k'} \mathcal{C}$ with $f;h' = g;k'$ there is a unique morphism $u : \mathcal{D} \to \mathcal{D}'$ such that $h' = h;u$ and $k' = k;u$. □

$$\begin{tikzcd} \mathcal{A} \arrow[r, "g"] \arrow[d, "f"'] & \mathcal{C} \arrow[d, "k"] \arrow[ddr, "k'"] & \\ \mathcal{B} \arrow[r, "h"] \arrow[drr, "h'"'] & \mathcal{D} \arrow[dr, dotted, "u"] & \\ & & \mathcal{D}' \end{tikzcd} \qquad \begin{tikzcd} \mathcal{A} \arrow[r, "Q"] \arrow[d, "P"'] & \mathcal{C} \arrow[d, "S"] \arrow[ddr, "S'"] & \\ \mathcal{B} \arrow[ur, "W"] \arrow[r, "R"] \arrow[drr, "R'"'] & \mathcal{D} \arrow[dr, dotted, "U"] & \\ & & \mathcal{D}' \end{tikzcd}$$

In the same way as pullbacks, pushouts are also unique up to isomorphism.

And, dual to the definition of categorical products, one obtains the definition of categorical sums (coproducts) as pushouts of (unique) morphisms with an initial object as source.

Although the relational characterisation of direct sums is obviously dual to that of direct products, too, a naïve relational dualisation of the tabulation conditions does *not* yield a concept that would give rise to pushouts as tabulations give rise to pullbacks.

The reason for this is that for a cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ of mappings, the relation $R;S^{\smile}$ is always difunctional, while for a span $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ the relation $P^{\smile};Q$ can essentially be arbitrary. (More precisely, in a tabular allegory there is such a span for every relation: its tabulation.)

We now abstract Kawahara's relational characterisation of pushouts of mappings [Kaw90, Thm. 3.1] away from the span $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$, and instead use the relation $P^{\smile};Q$ as our starting point. Given the rôle of pushouts in graph transformation to *glue* together parts of rules and parts of host graphs, we choose the name "*gluing*" for this concept:

**Definition 5.3.2** [←127, 146] In a Dedekind category $\mathbf{D}$, let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

The cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ in the Dedekind category $\mathbf{D}$ is a *direct gluing* for $W$ iff the following equations hold:

$$R;S^{\smile} = W^{\boxast} \qquad R;R^{\smile} = W^{\rhd} \qquad S;S^{\smile} = W^{\lhd} \qquad R^{\smile};R \sqcup S^{\smile};S = \mathbb{I}_{\mathcal{D}} \ . \qquad \square$$

The last equation implies univalence of $R$ and $S$, and the second and third imply totality.

Just as direct products are direct tabulations for universal relations, direct sums are direct gluings for empty relations. Because of the zero law, we do not get deviations from the original definition.

We again establish a generalised factorisation property:

**Lemma 5.3.3** [←105] In a Dedekind category $\mathbf{D}$, let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

If the cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ is a direct gluing for $W$, and if the cospan $\mathcal{B} \xrightarrow{R'} \mathcal{D}' \xleftarrow{S'} \mathcal{C}$ consists of mapping that satisfy the following condition:

$$W;S' \sqsubseteq R' \qquad \text{and} \qquad W^{\smile};R' \sqsubseteq S' \ ,$$

then $U : \mathcal{D} \to \mathcal{D}'$ with $U := R^{\smile};R' \sqcup S^{\smile};S'$ is a mapping such that $R' = R;U$ and $S' = S;U$.

**Proof**: Factorisation follows easily from the assumptions:

$$\begin{aligned} R;U &= R;R^{\smile};R' \sqcup R;S^{\smile};S' \ = \ W^{\rhd};R' \sqcup W^{\boxast};S' \ = \ R' \sqcup W;S' \ = \ R' \\ S;U &= S;R^{\smile};R' \sqcup S;S^{\smile};S' \ = \ (W^{\smile})^{\boxast};R' \sqcup (W^{\smile})^{\lhd};S' \ = \ W^{\smile};R' \sqcup S' \ = \ S' \end{aligned}$$

Univalence follows from factorisation and univalence of $R'$ and $S'$:

$$U^{\smile};U = (R'^{\smile};R \sqcup S'^{\smile};S);U = R'^{\smile};R;U \sqcup S'^{\smile};S;U = R'^{\smile};R' \sqcup S'^{\smile};S' \sqsubseteq \mathbb{I}$$

Totality uses the fourth gluing condition:

$$\begin{aligned} \mathsf{dom}\, U &= \mathsf{dom}\,(R^{\smile};R' \sqcup S^{\smile};S') \ = \ \mathsf{dom}\,(R^{\smile};R') \sqcup \mathsf{dom}\,(S^{\smile};S') \\ &= \mathsf{dom}\,(R^{\smile};\mathsf{dom}\, R') \sqcup \mathsf{dom}\,(S^{\smile};\mathsf{dom}\, S') \ = \ \mathsf{dom}\,(R^{\smile}) \sqcup \mathsf{dom}\,(S^{\smile}) \\ &= (\mathbb{I} \sqcap R^{\smile};R) \sqcup (\mathbb{I} \sqcap S^{\smile};S) \ = \ \mathbb{I} \sqcap (R^{\smile};R \sqcup S^{\smile};S) \ = \ \mathbb{I} \qquad \square \end{aligned}$$

Direct gluings are unique up to isomorphism, too:

**Theorem 5.3.4** In a Dedekind category $\mathbf{D}$, let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

If the cospans $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ and $\mathcal{B} \xrightarrow{R'} \mathcal{D}' \xleftarrow{S'} \mathcal{C}$ are both direct gluings for $W$, then there is a bijective mapping $U : \mathcal{D} \to \mathcal{D}'$ such that $R' = R;U$ and $S' = S;U$.



**Proof**: With the gluing conditions for $\mathcal{B} \xrightarrow{R'} \mathcal{D}' \xleftarrow{S'} \mathcal{C}$ we obtain:

$$W;S' \sqsubseteq W^{\boxast};S' = R';S'^{\smile};S' \sqsubseteq R' \qquad \text{and} \qquad W^{\smile};R' \sqsubseteq (W^{\smile})^{\boxast};R' = S';R'^{\smile};R' \sqsubseteq S' \ .$$

With Lemma 5.3.3 we then know that $U := R^{\smile};R' \sqcup S^{\smile};S'$ is a mapping that factorises $R'$ and $S'$.

By the same argument for $U^{\smile}$, we obtain that $U$ is also bijective. $\square$

And direct gluings are pushouts:

**Theorem 5.3.5** [←106] Let $\mathbf{D}$ be a Dedekind category, and let $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ be a span in $\mathsf{Map}\,\mathbf{D}$, that is, $P$ and $Q$ are mappings.

If the cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ in the Dedekind category $\mathbf{D}$ is a direct gluing for $W := P^{\smile};Q$, then it is a pushout for $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ in $\mathsf{Map}\,\mathbf{D}$.

**Proof**: The gluing properties imply that $R$ and $S$ are mappings. For commutativity, we first show one inclusion:

$$P;R \sqsupseteq P;R;\mathsf{ran}\, S = P;R;S^{\smile};S = P;(P^{\smile};Q)^{\boxast};S \sqsupseteq P;P^{\smile};Q;S \sqsupseteq Q;S$$

The opposite inclusion is derived in the same way, so we have equality.

Now assume another cospan $\mathcal{B} \xrightarrow{R'} \mathcal{D}' \xleftarrow{S'} \mathcal{C}$ in $\mathsf{Map}\,\mathbf{D}$ such that $P;R' = Q;S'$. This commutativity together with univalence of $P$ and $Q$ implies

$$W;S' = P^{\smile};Q;S' = P^{\smile};P;R' \sqsubseteq R' \qquad \text{and} \qquad W^{\smile};R' = Q^{\smile};P;R' = Q^{\smile};Q;S' \sqsubseteq S' \ .$$

With Lemma 5.3.3 we then know that $U := R^{\smile};R' \sqcup S^{\smile};S'$ is a mapping that factorises $R'$ and $S'$. So we only have to show that $U$ is uniquely determined. Assume $U' : \mathcal{D} \to \mathcal{D}'$ with $R;U' = R'$ and $S;U' = S'$. Then:

$$U' = (R^{\smile};R \sqcup S^{\smile};S);U' = R^{\smile};R;U' \sqcup S^{\smile};S;U' = R^{\smile};R' \sqcup S^{\smile};S' = U \qquad \square$$

As for pullbacks, we now reconstruct the set-theoretic pushout construction in relational terms. Here, we need a direct sum, which involves working with joins and thus requires the setting of a distributive allegory. Furthermore, we need a quotient, and for defining the equivalence relation for the quotient construction we need equivalence closure (which we formulate using reflexive transitive closure). Therefore, we need a locally complete distributive allegory, which for us means a Dedekind category setting.

**Definition 5.3.6** [←106] Let $\mathbf{D}$ be a Dedekind category, and let $W : \mathcal{B} \leftrightarrow \mathcal{C}$ be an arbitrary relation.

If $(\mathcal{S}, \iota, \kappa)$ is a direct sum for $\mathcal{B}$ and $\mathcal{C}$, then define:

$$V := \iota^{\smile};W;\kappa \ , \qquad \Theta := (V \sqcup V^{\smile})^{*}$$

If there exists a quotient $(\mathcal{D}, \theta)$ for $\Theta$, then the cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ is called a *constructed gluing for* $W$, where $R := \iota;\theta$ and $S := \kappa;\theta$. $\square$

Well-definedness is shown easily:

**Theorem 5.3.7** If $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ is a constructed gluing for $W : \mathcal{B} \leftrightarrow \mathcal{C}$, then it is a direct gluing for $W$.

**Proof**: We first show the fourth equation:

$$R^{\smile};R \sqcup S^{\smile};S = \theta^{\smile};\iota;\iota^{\smile};\theta \sqcup \theta^{\smile};\kappa;\kappa^{\smile};\theta = \theta^{\smile};(\iota;\iota^{\smile} \sqcup \kappa;\kappa^{\smile});\theta = \theta^{\smile};\theta = \mathbb{I}$$

From the definition of $\Theta$ it is easy to see that we have the following decomposition:

$$\Theta = \iota^{\smile};W^{\rhd};\iota \sqcup \kappa^{\smile};W^{\lhd};\kappa \sqcup \iota^{\smile};W^{\boxast};\kappa \sqcup \kappa^{\smile};(W^{\boxast})^{\smile};\iota$$

From this, the remaining three equations follow directly:

$$\begin{aligned} R;S^{\smile} &= \iota;\theta;\theta^{\smile};\kappa^{\smile} &= \iota;\Theta;\kappa^{\smile} &= W^{\boxast} \\ R;R^{\smile} &= \iota;\theta;\theta^{\smile};\iota^{\smile} &= \iota;\Theta;\iota^{\smile} &= W^{\rhd} \\ Q;Q^{\smile} &= \kappa;\theta;\theta^{\smile};\kappa^{\smile} &= \kappa;\Theta;\kappa^{\smile} &= W^{\lhd} \end{aligned}$$

□

As in the case of pullbacks, this construction involves an intermediate object, here the direct sum for $\mathcal{B}$ and $\mathcal{C}$, which need not exist even if there is a direct gluing in $\mathbf{D}$. And it is also possible that there are categorical pushouts in $\mathsf{Map}\,\mathbf{D}$ which are not direct gluings, but this kind of setting is not relevant for our investigations.

A simple corollary of the relational view of pushouts is the well-known and important fact that pushouts "preserve monos"; here we show that they preserve the relational property of injectivity (which is slightly stronger than monicity, even in its restriction to mappings):

**Lemma 5.3.8** [←113] Let a Dedekind category $\mathbf{D}$ and a span $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ in $\mathsf{Map}\,\mathbf{D}$ be given. If the cospan $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$ in $\mathsf{Map}\,\mathbf{D}$ is a direct gluing for $P^{\smile};Q$, and if $Q$ is injective, then so is $R$. (And, analogously, if $P$ is injective, then so is $S$.)

**Proof**: By a gluing property, injectivity of $Q$, and univalence of $P$:

$$R;R^{\smile} = (P^{\smile};Q)^{\rhd} = (P^{\smile};Q;Q^{\smile};P)^{*} \sqsubseteq (P^{\smile};P)^{*} \sqsubseteq \mathbb{I}$$

Injectivity of $S$ follows in the same way from injectivity of $P$. □

Summarising the relation between direct gluings in a Dedekind category and pushouts in its category of mappings, we have:

**Theorem 5.3.9** In a Dedekind category $\mathbf{D}$, let $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ be a span in $\mathsf{Map}\,\mathbf{D}$.

If there is a direct gluing for $P^{\smile};Q$, then the categorical pushouts in $\mathsf{Map}\,\mathbf{D}$ for $\mathcal{B} \xleftarrow{P} \mathcal{A} \xrightarrow{Q} \mathcal{C}$ and the direct gluings in $\mathbf{D}$ for $P;Q^{\smile}$ coincide.

**Proof**: If there is a direct gluing $\mathcal{B} \xrightarrow{R} \mathcal{D} \xleftarrow{S} \mathcal{C}$, then by Theorem 5.3.5 it is also a pushout. Since pushouts are unique up to isomorphism, all other pushouts are isomorphic to it. □

Therefore, if the construction of Def. 5.3.6 is always possible, then it creates all pushouts (up to isomorphism):

**Corollary 5.3.10** If the Dedekind category $\mathbf{D}$ has all direct sums and quotients, then the categorical pushouts in $\mathsf{Map}\,\mathbf{D}$ are direct gluings. □



## 5.4 Pushout Complements

The problem of constructing pushout complements arises in the double-pushout approach to graph rewriting, where the left-hand side of a rule, a morphism $\Phi$ from a gluing object $\mathcal{G}$ to a left-hand side object $\mathcal{L}$ and a matching morphism $\Phi$ from $\mathcal{L}$ into an application graph $\mathcal{A}$ are given, and a host graph $\mathcal{H}$ together with morphisms $\Xi : \mathcal{G} \to \mathcal{H}$ and $\Psi : \mathcal{H} \to \mathcal{A}$ needs to be constructed in such a way that the resulting diagram is a pushout.

$$\begin{tikzcd} \mathcal{L} \arrow[d, "\mathrm{X}"'] & \mathcal{G} \arrow[l, "\Phi"'] \arrow[d, "\Xi"] \\ \mathcal{A} & \mathcal{H} \arrow[l, "\Psi"] \end{tikzcd}$$

This is a problem since such a *pushout complement* $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ does not exist for all constellations $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$

In the category of graphs, the *gluing condition* of Def. 1.2.3 is a necessary and sufficient condition for the existence of pushout complements. Although the node-and-edges-level formulation of Def. 1.2.3 is standard in the literature on categoric graph transformation, it is very much "against the style" of the categoric approach. Kawahara's component-free formulation [Kaw90] employs an embedding of relational calculus in topos theory, so, apart from notation, there are also some minor technical differences to the Dedekind category setting we are using here. On the whole, however, the material of [Kaw90] is easily translated into the present context.

Let us first state a relational variant of the gluing condition, using essentially Kawahara's identification condition, but a different formulation of the dangling condition, and an additional condition that is important in the single-pushout approach:

**Definition 5.4.1** Let two relations $\Phi : \mathcal{G} \leftrightarrow \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \leftrightarrow \mathcal{A}$ in a strict Dedekind category $\mathbf{D}$ be given.

- We say that the *identification condition* holds iff X is almost-injective besides $\mathsf{ran}\,\Phi$:

$$\mathrm{X};\mathrm{X}^{\smile} \sqsubseteq \mathbb{I} \sqcup (\mathsf{ran}\,\Phi);\mathrm{X};\mathrm{X}^{\smile};\mathsf{ran}\,\Phi\ .$$

- We say that the *dangling condition* holds iff

$$\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\sim} \sqsubseteq (\mathsf{ran}\,\Phi);\mathrm{X}$$

- We call $\mathrm{X} : \mathcal{L} \to \mathcal{A}$ is called *conflict-free for* $\Phi$ iff $\mathsf{ran}\,(\Phi;\mathrm{X};\mathrm{X}^{\smile}) \sqsubseteq \mathsf{ran}\,\Phi$. □

Kawahara, whose framework does not provide semi-complements, formulates the dangling condition via a pseudo-complement:

$$\mathsf{ran}\,\mathrm{X} \sqcup (\mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X})) = \mathbb{I}$$

We consider our formulation using a semi-complement as easier to relate with the original definition, and also as easier to use in proofs. It is, however, equivalent to Kawahara's definition, when put into the context of the gluing definition, and even in the weaker context of conflict-freeness:

**Lemma 5.4.2** (i) The identification condition implies that X is conflict-free for $\Phi$.
(ii) If X is conflict-free for $\Phi$, then

$$\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\smile} \sqsubseteq (\mathsf{ran}\,\Phi);\mathrm{X} \qquad \text{iff} \qquad \mathsf{ran}\,\mathrm{X} \sqcup (\mathsf{ran}\,\mathrm{X} \rightarrow \mathsf{ran}\,(\Phi;\mathrm{X})) = \mathbb{I}$$

**Proof**:

(i) Obvious.

(ii) Via the definition of semi-complements, Kawahara's formulation is equivalent to

$$(\mathsf{ran}\,\mathrm{X})^{\smile} \sqsubseteq \mathsf{ran}\,\mathrm{X} \rightarrow \mathsf{ran}\,(\Phi;\mathrm{X}) \ ,$$

and via the definition of pseudo-complements this is equivalent to

$$(\mathsf{ran}\,\mathrm{X})^{\smile} \sqcap \mathsf{ran}\,\mathrm{X} \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \ .$$

With A.2.2.ii) and (iii), this transforms into

$$\mathsf{ran}\,(\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\smile}) \sqsubseteq \mathsf{ran}\,((\mathsf{ran}\,\Phi);\mathrm{X}) \ ,$$

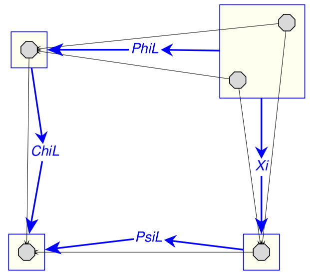
which is obviously implied by our formulation; the opposite implication may be shown as follows:

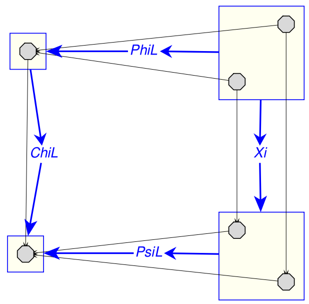
$$\begin{array}{rcll}
\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\smile} & = & \mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\smile};\mathrm{X}^{\smile};\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\smile} & \text{univalence of X} \\
& = & \mathrm{X};\mathsf{ran}\,(\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\smile}) & \text{univalence of X} \\
& \sqsubseteq & \mathrm{X};\mathsf{ran}\,((\mathsf{ran}\,\Phi);\mathrm{X}) & \text{assumption} \\
& = & \mathrm{X};\mathrm{X}^{\smile};(\mathsf{ran}\,\Phi);\mathrm{X} & \text{univalence of X} \\
& = & (\mathsf{ran}\,\Phi);\mathrm{X};\mathrm{X}^{\smile};(\mathsf{ran}\,\Phi);\mathrm{X} & \text{conflict-free} \\
& = & (\mathsf{ran}\,\Phi);\mathrm{X} & \text{univalence of X}
\end{array}$$
□

It is well known that the pushout complement is uniquely determined (up to isomorphism) if $\Phi$ is injective. In general, it is possible that different, non-isomorphic pushout complements exist. The construction given by Kawahara as part of [Kaw90, Thm. 3.6] builds a pushout complement which is the subobject of $\mathcal{A}$ for $\mathsf{ran}\,\mathrm{X} \rightarrow \mathsf{ran}\,(\Phi;\mathrm{X})$, so even in ambiguous cases, where $\Phi$ is not injective, the morphism $\Psi$ from host to application graph constructed in this way will be injective. As we shall see below, an equivalent definition of this subobject uses $(\mathsf{ran}\,\mathrm{X})^{\smile} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X})$, which is perhaps easier to understand intuitively: The whole context $(\mathsf{ran}\,\mathrm{X})^{\smile}$ of the image of the left-hand side is preserved, and the image $\mathsf{ran}\,(\Phi;\mathrm{X})$ of the gluing graph $\mathcal{G}$.

A further alternative, which would construct a different pushout complement in cases where $\Phi$ is not injective, would not extract the image of $\mathcal{G}$ from $\mathcal{A}$, but rather glue $\mathcal{G}$ directly with the context $(\mathsf{ran}\,\mathrm{X})^{\smile}$ and then transfer identification induced by X, but not transfer identifications induced only by $\Phi$. In the following drawing, this is shown in the right part for a two-node discrete gluing graph $\mathcal{G}$, while the standard pushout complement is in the left part:

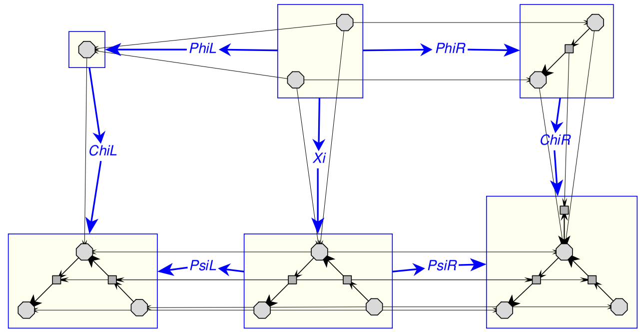




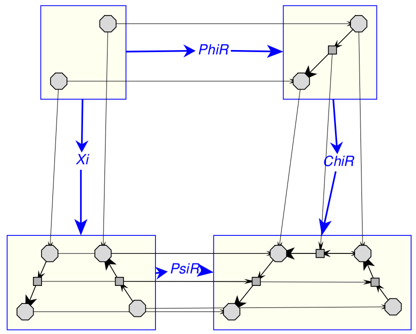



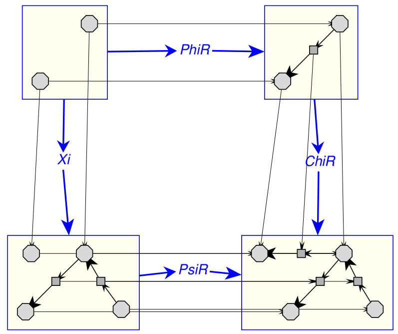

If the application graph $\mathcal{A}$ contains edges incident to such identified nodes, then these edges may be redirected to any of the interface nodes in the gluing graphs, and therefore this under-determination gives rise to significantly different results.

In the following example application, the edge inserted by the rule's right-hand side turns into a loop attached to the identified redex node under the standard pushout complement construction:




Alternative pushout complements do not identify the interface nodes, and may attach the edges incident to the redex node in different ways to the interface nodes in the host graphs, which the gives rise to quite different results:



For this reason, double-pushout graph rewriting usually insists on injective left-hand side morphisms of rules (see also [HMP00]).

In addition we shall see below (Theorem 5.4.11) that the standard construction is also useful in other contexts, and it is also much simpler to formulate, so we stick with it.

In this context, the following property is helpful:

**Lemma 5.4.3** [←110] [Kaw90, Lemma 3.3] In a Heyting algebra (i.e., in a pseudocomplemented lattice) with maximum element $\top$, let $A$ and $B$ be elements such that $B \leqslant A$. Then there exists an element $C$ satisfying $A \vee C = \top$ and $A \wedge C = B$ iff $A \vee (A \to B) = \top$. When this is the case, then $C = A \to B$.

**Proof**: "$\Rightarrow$": Assume that there exists an element $C$ satisfying $A \vee C = \top$ and $A \wedge C = B$.

With the definition of relative pseudo-complements, $A \wedge C \sqsubseteq B$ is equivalent to $C \sqsubseteq A \to B$. Then $\top = A \vee C \sqsubseteq A \vee (A \to B)$, so that the latter is equal to $\top$, too. Together with $A \wedge C = B = A \wedge (A \to B)$ this implies $A \to B = C$:

$$\begin{aligned} A \to B &= \top \sqcap (A \to B) = (A \vee C) \sqcap (A \to B) \\ &= (A \sqcap (A \to B)) \vee (C \sqcap (A \to B)) = B \vee C = C \end{aligned}$$

"$\Leftarrow$" is trivial. □

It allows to show that the range of $\Psi$ is completely determined by $\Phi$ and $\mathrm{X}$:

**Lemma 5.4.4** [←113] [Kaw90, Cor. 3.2 (PO.4)] If $\mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A} \xleftarrow{\Psi} \mathcal{H}$ is a pushout for $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ in $\mathsf{Map}\,\mathbf{D}$ for a Dedekind category $\mathbf{D}$, then $\mathsf{ran}\,\Psi = \mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X})$.

**Proof**:
$$\begin{aligned} \mathsf{ran}\,\Psi \sqcap \mathsf{ran}\,\mathrm{X} &= \Psi^{\smile};\Psi \sqcap \mathrm{X}^{\smile};\mathrm{X} \\ &= \mathrm{X}^{\smile};(\mathrm{X};\Psi^{\smile};\Psi \sqcap \mathrm{X}) && \mathrm{X}\ \text{univalent} \\ &= \mathrm{X}^{\smile};((\Phi^{\smile};\Xi)^{\boxast};\Psi \sqcap \mathrm{X}) \\ &\sqsubseteq \mathrm{X}^{\smile};\mathsf{ran}\,\Phi;\mathrm{X} \\ &= \mathsf{ran}\,(\Phi;\mathrm{X}) \end{aligned}$$

Since the opposite inclusion follows from commutativity, we have $\mathsf{ran}\,\Psi \sqcap \mathsf{ran}\,\mathrm{X} = \mathsf{ran}\,(\Phi;\mathrm{X})$.

Since the gluing properties imply $\mathsf{ran}\,\Psi \sqcup \mathsf{ran}\,\mathrm{X} = \mathbb{I}$, we can apply 5.4.3 to the Heyting algebra of partial identities on $\mathcal{H}$, and obtain $\mathsf{ran}\,\Psi = \mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X})$. □

As mentioned above, we also have an alternative formulation for that range:

**Lemma 5.4.5** Under the gluing condition, we have

$$\mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X}) \quad = \quad (\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X}) \ .$$



**Proof**:
$$\begin{aligned} & (\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X}) \sqsubseteq \mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X}) \\ \Leftrightarrow\ & ((\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X})) \sqcap \mathsf{ran}\,\mathrm{X} \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \\ \Leftrightarrow\ & ((\mathsf{ran}\,\mathrm{X})^{\sim} \sqcap \mathsf{ran}\,\mathrm{X}) \sqcup \mathsf{ran}\,(\Phi;\mathrm{X}) \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \\ \Leftrightarrow\ & (\mathsf{ran}\,\mathrm{X})^{\sim} \sqcap \mathsf{ran}\,\mathrm{X} \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \\ \Leftrightarrow\ & \mathsf{ran}\,\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\sim} \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \\ \Leftrightarrow\ & \mathsf{ran}\,(\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\sim}) \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \\ \Leftarrow\ & \mathsf{ran}\,((\mathsf{ran}\,\Phi);\mathrm{X}) \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) && \text{dangling} \\ \Leftrightarrow\ & \mathsf{ran}\,(\Phi;\mathrm{X}) \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \end{aligned}$$

The opposite inclusion follows from Lemma 2.7.10. □

In the presence of the gluing condition, there is therefore no difference between the two formulations. In the absence of the gluing condition, the relative pseudo-complement deletes dangling edges, while the semi-complement preserves them together with the nodes they are attached to. We transfer Kawahara's construction into our setting:

**Definition 5.4.6** [←115, 130, 138, 139, 150] Let two relations $\Phi : \mathcal{G} \leftrightarrow \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \leftrightarrow \mathcal{A}$ in a strict Dedekind category $\mathbf{D}$ be given.

If $q : \mathsf{PId}\,\mathcal{A}$ is a partial identity on $\mathcal{A}$, then a *subobject host construction for* $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$ *by* $q$ is a diagram $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ where $\Psi$ is a subobject injection for $q$ and $\Xi := \Phi;\mathrm{X};\Psi^{\smile}$.

A *straight host construction* is a subobject host construction by $\mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X})$.

A *sloppy host construction* is a subobject host construction by $(\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X})$. □

$$\begin{tikzcd} \mathcal{L} \arrow[d, "\mathrm{X}"'] & \mathcal{G} \arrow[l, "\Phi"'] \arrow[d, "\Xi"] \\ \mathcal{A} & \mathcal{H} \arrow[l, "\Psi"] \end{tikzcd}$$

Since we are going to employ the straight host construction in different settings, we first show a few general properties:

**Lemma 5.4.7** [←112, 113, 115, 179, 181, 199] If $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ is a subobject host construction for $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$ by $q : \mathsf{PId}\,\mathcal{A}$ with $\mathsf{ran}\,(\Phi;\mathrm{X}) \sqsubseteq q$, then the following properties hold:

(i) $\Phi^{\smile};\Xi;\Xi^{\smile};\Phi = \Phi^{\smile};\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};\Phi$

(ii) If $\Phi$ is univalent, then $(\Phi^{\smile};\Xi)^{\rhd} = (\mathsf{ran}\,\Phi;\mathrm{X})^{\rhd}$

(iii) If $\Phi$ is difunctional and $\Phi;\mathrm{X}$ is univalent, then $\Xi^{\smile};\Phi;\Phi^{\smile};\Xi \sqsubseteq \mathbb{I}$ and therefore also $(\Phi^{\smile};\Xi)^{\lhd} = \mathbb{I}$ and

$$(\Phi^{\smile};\Xi)^{\boxast} = \Phi^{\smile};\Xi = \Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile} = \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} \ .$$

(iv) If $\Phi;\mathrm{X}$ is univalent, then $\Xi$ is univalent.

(v) If $\Phi;\mathrm{X}$ is total, then $\Xi$ is total.

(vi) For a straight host construction only: If X is conflict-free on $\mathsf{ran}\,\Phi$, then $\mathrm{X};\Psi^{\smile} = \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile}$.

**Proof**:

(i)
$$\begin{aligned}\Phi^{\smile};\Xi;\Xi^{\smile};\Phi &= \Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile};\Psi;\mathrm{X}^{\smile};\Phi^{\smile};\Phi\\ &= \Phi^{\smile};\Phi;\mathrm{X};\mathsf{ran}\,\Psi;\mathrm{X}^{\smile};\Phi^{\smile};\Phi \;=\; \Phi^{\smile};\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};\Phi\end{aligned}$$

(ii) follows from (i).

(iii) $\Xi^{\smile};\Phi;\Phi^{\smile};\Xi = \Psi;\mathrm{X}^{\smile};\Phi^{\smile};\Phi;\Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile} \sqsubseteq \Psi;\mathrm{X}^{\smile};\Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile} \sqsubseteq \Psi;\Psi^{\smile} = \mathbb{I}$

(iv) $\Xi^{\smile};\Xi = \Psi;\mathrm{X}^{\smile};\Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile} \sqsubseteq \Psi;\Psi^{\smile} = \mathbb{I}$

(v) $\mathsf{dom}\,\Xi = \mathsf{dom}\,(\Phi;\mathrm{X};\Psi^{\smile}) = \mathsf{dom}\,(\Phi;\mathrm{X};\mathsf{ran}\,\Psi) = \mathsf{dom}\,(\Phi;\mathrm{X}) = \mathbb{I}$

(vi) By definition of $\Psi$ in a straight host construction we have

$$\mathsf{ran}\,\Psi \sqcap \mathsf{ran}\,\mathrm{X} \sqsubseteq \mathsf{ran}\,(\Phi;\mathrm{X}) \ .$$

Therefore, the fact that X is conflict-free on $\mathsf{ran}\,\Phi$ gives us

$$\begin{aligned}\mathrm{X};\Psi^{\smile} &= \mathrm{X};\mathsf{ran}\,(\Phi;\mathrm{X});\Psi^{\smile}\\ &= \mathrm{X};\mathrm{X}^{\smile};\mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile}\\ &= \mathsf{ran}\,\Phi;\mathrm{X};\mathrm{X}^{\smile};\mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile}\\ &= \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} \ .\end{aligned}$$
□

In a setting where all pushouts of mappings are gluings, we have:

**Theorem 5.4.8 (Pushout Complement)** [Kaw90, Thm. 3.6] If $\Phi : \mathcal{G} \to \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \to \mathcal{A}$ are two mappings in a strict Dedekind category $\mathbf{D}$, such that the gluing condition holds, then a straight host construction $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ for $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$ is a pushout complement for $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$ in $\mathsf{Map}\,\mathbf{D}$.

**Proof**: We first show that $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ is a pushout complement, that is, we have to show that $\mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A} \xleftarrow{\Psi} \mathcal{H}$ is a gluing for $W := \Phi^{\smile};\Xi$.

With Lemma 5.4.7.iv) we have univalence of $\Xi$, which implies that $\Phi^{\smile};\Xi;\Xi^{\smile};\Phi$ is idempotent, so we have with Lemma 5.4.7.i):

$$\begin{aligned}(\Phi^{\smile};\Xi;\Xi^{\smile};\Phi)^{+} &= \Phi^{\smile};\Xi;\Xi^{\smile};\Phi \;=\; \mathsf{ran}\,\Phi;\mathrm{X};\mathrm{X}^{\smile};\mathsf{ran}\,\Phi\\ (\Phi^{\smile};\Xi;\Xi^{\smile};\Phi)^{*} &= \mathbb{I} \sqcup \mathsf{ran}\,\Phi;\mathrm{X};\mathrm{X}^{\smile};\mathsf{ran}\,\Phi \;=\; \mathrm{X};\mathrm{X}^{\smile}\end{aligned}$$

The last equation follows from the identification condition; therefore, one of the gluing properties is already shown:

$$\mathrm{X};\mathrm{X}^{\smile} = (\Phi^{\smile};\Xi;\Xi^{\smile};\Phi)^{*} = (\Phi^{\smile};\Xi)^{\triangleright\!\!\triangleright} = W^{\triangleright\!\!\triangleright}$$



Lemma 5.4.7.iii) shows $\Psi;\Psi^{\smile} = \mathbb{I} = (\Xi^{\smile};\Phi;\Phi^{\smile};\Xi)^{*} = (\Phi^{\smile};\Xi)^{\triangleleft\!\!\triangleleft} = W^{\triangleleft\!\!\triangleleft}$. For the mixed composition we use Lemma 5.4.7.vi) and (iii):

$$\mathrm{X};\Psi^{\smile} = \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} = (\Phi^{\smile};\Xi)^{\boxtimes} = W^{\boxtimes}$$

Finally, we also have ordinary commutativity:

$$\Xi;\Psi = \Phi;\mathrm{X};\Psi^{\smile};\Psi = \Phi;\mathrm{X};(\mathsf{ran}\,\mathrm{X} \to \mathsf{ran}\,(\Phi;\mathrm{X})) = \Phi;\mathrm{X} \ .$$
□

Obviously, even where $\Phi$ is not injective, this construction delivers a pushout complement with injective $\Psi$. One of the consequences is that a simplified variant $\mathrm{X};\Psi^{\smile} = \Phi^{\smile};\Xi$ of "alternative commutativity" holds, as can be seen via (iii).

For completeness' sake, we show why the pushout complement is uniquely determined when $\Phi$ is injective (this is missing in [Kaw90]):

**Proposition 5.4.9** If $\Phi$ is injective, then there is only one pushout complement up to isomorphism.

**Proof**: Let $\Phi : \mathcal{G} \to \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \to \mathcal{A}$ be two mappings in a strict Dedekind category $\mathbf{D}$, and assume that $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ and $\mathcal{G} \xrightarrow{\Xi'} \mathcal{H}' \xrightarrow{\Psi'} \mathcal{A}$ are two pushout complements for $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$. Define:

$$\mathrm{Y} := \Psi;\Psi'^{\smile}$$

Since 5.4.4 implies that $\mathsf{ran}\,\Psi = \mathsf{ran}\,\Psi'$, and since with 5.3.8, $\Psi$ and $\Psi'$ are injective, we obtain

$$\mathrm{Y}^{\smile};\mathrm{Y} = \Psi';\Psi^{\smile};\Psi;\Psi'^{\smile} = \Psi';\mathsf{ran}\,\Psi;\Psi'^{\smile} = \Psi';\Psi'^{\smile} = \mathbb{I} \ ,$$

and in the same way also $\mathrm{Y};\mathrm{Y}^{\smile} = \mathbb{I}$, so Y is an isomorphism. Factorisation follows easily:

$$\begin{aligned}\mathrm{Y};\Psi' &= \Psi;\Psi'^{\smile};\Psi' \;=\; \Psi;\mathsf{ran}\,\Psi' \;=\; \Psi\\ \Xi;\mathrm{Y} &= \Xi;\Psi;\Psi'^{\smile} \;=\; \Phi;\mathrm{X};\Psi'^{\smile} \;=\; \Xi';\Psi';\Psi'^{\smile} \;=\; \Xi';W^{\triangleleft\!\!\triangleleft}\\ &= \Xi';(\Xi'^{\smile};\Phi;\Phi^{\smile};\Xi')^{*}\\ &= \Xi';(\Xi'^{\smile};\Xi')^{*} \;=\; \Xi' \qquad \Phi \text{ total and injective}\end{aligned}$$
□

Even if the gluing condition does not hold, our host constructions still produce commuting squares in $\mathsf{Map}\,\mathbf{D}$:

**Theorem 5.4.10** If $\Phi : \mathcal{G} \to \mathcal{L}$ and $\mathrm{X} : \mathcal{L} \to \mathcal{A}$ are two mappings in a strict Dedekind category $\mathbf{D}$, and $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{A}$ is a straight or sloppy host construction for $\mathcal{G} \xrightarrow{\Phi} \mathcal{L} \xrightarrow{\mathrm{X}} \mathcal{A}$, then $\Xi$ and $\Psi$ are mappings, and $\Phi;\mathrm{X} = \Xi;\Psi$.

**Proof**: $\Psi$ is a mapping by construction, and $\Xi$ by Lemma 5.4.7.iv) and Lemma 5.4.7.v). Commutativity holds by definition of $\Xi$ since $\mathsf{ran}\,(\Phi;\mathrm{X}) \sqsubseteq \mathsf{ran}\,\Psi$:

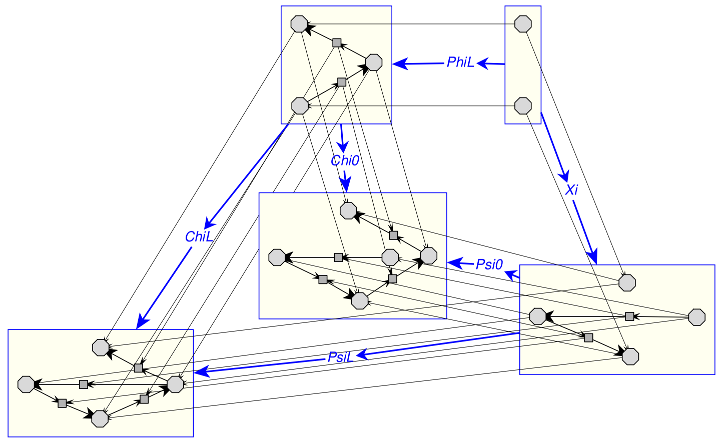
$$\Xi;\Psi = \Phi;\mathrm{X};\Psi^{\smile};\Psi = \Phi;\mathrm{X}$$
□

Because of the pushout property, the pushout for $\mathcal{L} \xleftarrow{\Phi} \mathcal{G} \xrightarrow{\Xi}$ (if it exists) then factors this square via a uniquely determined mapping $Y$.

This shows that the straight and sloppy host constructions *without the gluing condition* give rise to the left-hand side of "restricting derivations" (see 1.2.5): a pushout square with an additional morphism tacked onto its tip:

$$
\begin{tikzcd}
 & \mathcal{L} \arrow[dl, "X"'] \arrow[d, "X_0"] & \mathcal{G} \arrow[l, "\Phi"'] \arrow[d, "\Xi"] \\
\mathcal{A} & \mathcal{A}_0 \arrow[l, "Y"'] & \mathcal{H} \arrow[l, "\Psi_0"']
\end{tikzcd}
$$

An example for a sloppy host is the following, shown together with the resulting pushout:




For implementing restricting derivations, the sloppy host construction can be a useful alternative to the straight host construction, representing a slightly more careful approach that does not perform implicit deletions.

Certain restricting derivation steps implement single-pushout derivation steps. The single-pushout approach uses a single pushout in a category of partial graph morphisms as its rewriting step, see 1.2.6. For a node-and-edges-level formulation of conflict-freeness it is well-known that the induced single-pushout squares have a total embedding of the right-hand side into the application graph [Löw90, Cor. 3.18.5]. In a restricting derivation step, we always have a total right-hand side morphism, so we need conflict-freeness to obtain a pushout of partial functions with a restricting derivation. With our component-free definition of conflict-freeness we can prove this result for arbitrary Dedekind categories:

**Theorem 5.4.11** [←110, 130] In a Dedekind category **D**, let three mappings in the constellation $\mathcal{A} \xleftarrow{X_L} \mathcal{L} \xleftarrow{\Phi_L} \mathcal{G} \xrightarrow{\Phi_R} \mathcal{R}$ be given.

If $X_L$ is conflict-free on $\mathsf{ran}\,\Phi_L$, then a straight host construction of $\mathcal{A} \xleftarrow{\Psi_L} \mathcal{H} \xleftarrow{\Xi} \mathcal{G}$ for $\mathcal{A} \xleftarrow{X_L} \mathcal{L} \xleftarrow{\Phi_L} \mathcal{G}$ followed by a pushout $\mathcal{R} \xrightarrow{X_R} \mathcal{B} \xleftarrow{\Psi_R} \mathcal{H}$ for $\mathcal{R} \xleftarrow{\Phi_R} \mathcal{G} \xrightarrow{\Xi} \mathcal{H}$ in $\mathsf{Map}\,\mathbf{D}$ yields a pushout in $\mathsf{Pfn}\,\mathbf{D}$.



**Proof**: Define: $\Phi := \Phi_L^{\smile};\Phi_R$ and $\Psi := \Psi_L^{\smile};\Psi_R$. Then we have commutativity:

$$
\begin{array}{rcll}
\Phi;X_R & = & \Phi_L^{\smile};\Phi_R;X_R & \\
 & = & \Phi_L^{\smile};\Xi;\Psi_R & \text{commutativity of rhs-pushout} \\
 & = & \Phi_L^{\smile};\Phi_L;X_L;\Psi_L^{\smile};\Psi_R & \text{Def. } \Xi \text{ in Def. 5.4.6} \\
 & = & \mathsf{ran}\,\Phi_L;X_L;\Psi_L^{\smile};\Psi_R & \Phi_L \text{ univalent} \\
 & = & X_L;\Psi_L^{\smile};\Psi_R & \text{Lemma 5.4.7.vi)} \\
 & = & X_L;\Psi &
\end{array}
$$

As a preparation for the remainder of the proof, let us first reformulate the setup of the right-hand-side pushout:

$$
\begin{array}{rcl}
\Phi_R^{\smile};\Xi & = & \Phi_R^{\smile};\Phi_L;X_L;\Psi_L^{\smile} \;=\; \Phi^{\smile};X_L;\Psi_L^{\smile} \\
\Phi_R^{\smile};\Xi;\Xi^{\smile};\Phi_R & = & \Phi^{\smile};X_L;\Psi_L^{\smile};\Psi_L;X_L^{\smile};\Phi \;=\; \Phi^{\smile};X_L;X_L^{\smile};\Phi
\end{array}
$$

For the last equality, we used $\mathsf{ran}\,(\Phi^{\smile};X_L) = \mathsf{ran}\,(\Phi_L;X_L) \sqsubseteq \mathsf{ran}\,\Psi_L$. Now we can use this to simplify the gluing properties of the right-hand side pushout:

$$
\begin{array}{rcl}
X_R;\Psi^{\smile} & = & X_R;\Psi_R^{\smile};\Psi_L \;=\; (\Phi_R^{\smile};\Xi)^{\boxast};\Psi_L \;=\; (\Phi_R^{\smile};\Xi;\Xi^{\smile};\Phi_R)^*;\Phi_R^{\smile};\Xi;\Psi_L \\
 & = & (\Phi^{\smile};X_L;X_L^{\smile};\Phi)^*;\Phi^{\smile};X_L;\Psi_L^{\smile};\Psi_L \;=\; (\Phi^{\smile};X_L;X_L^{\smile};\Phi)^*;\Phi^{\smile};X_L \;=\; (\Phi^{\smile};X_L)^{\boxast} \\
X_R;X_R^{\smile} & = & (\Phi_R^{\smile};\Xi)^{\rhd} \;=\; (\Phi_R^{\smile};\Xi;\Xi^{\smile};\Phi_R)^* \;=\; (\Phi^{\smile};X_L;X_L^{\smile};\Phi)^* \;=\; (\Phi^{\smile};X_L)^{\rhd} \\
\Psi;\Psi^{\smile} & = & \Psi_L^{\smile};\Psi_R;\Psi_R^{\smile};\Psi_L \;=\; \Psi_L^{\smile};(\Phi_R^{\smile};\Xi)^{\lhd};\Psi_L \;=\; \Psi_L^{\smile};\Phi_R^{\smile};\Xi;(\Phi_R^{\smile};\Xi)^{\boxast};\Psi_L \sqcup \Psi_L^{\smile};\Psi_L \\
 & = & \Psi_L^{\smile};\Psi_L;X_L^{\smile};\Phi;(\Phi^{\smile};X_L)^{\boxast} \sqcup \Psi_L^{\smile};\Psi_L \;=\; \Psi_L^{\smile};\Psi_L;(\Phi^{\smile};X_L)^{\lhd}
\end{array}
$$

For showing the pushout property, assume the existence of an object $\mathcal{B}'$ and two partial functions $\Psi' : \mathcal{A} \nrightarrow \mathcal{B}'$ and $X_R' : \mathcal{R} \nrightarrow \mathcal{B}'$ such that $\Phi;X_R' = X_L;\Psi'$. Then we define:

$$Y := X_R^{\smile};X_R' \sqcup \Psi^{\smile};\Psi'$$

$$
\begin{tikzcd}
 & \mathcal{L} \arrow[dl, "X_L"'] & \mathcal{G} \arrow[l, "\Phi_L"'] \arrow[r, "\Phi_R"] \arrow[d, "\Xi"] & \mathcal{R} \arrow[d, "X_R"] \arrow[ddr, "X_R'"] & \\
\mathcal{A} \arrow[drrrr, bend right, "\Psi'"] & & \mathcal{H} \arrow[ll, "\Psi_L"'] \arrow[r, "\Psi_R"] & \mathcal{B} \arrow[dr, dotted, "Y"'] & \\
 & & & & \mathcal{B}'
\end{tikzcd}
$$

We have:

$$
\begin{array}{rcll}
\mathsf{ran}\,X_L \sqcap \mathsf{dom}\,\Psi' & = & \mathsf{dom}\,((\mathsf{ran}\,X_L);\Psi') & \\
 & \sqsubseteq & \mathsf{dom}\,(X_L^{\smile};X_L;\Psi') & \\
 & = & \mathsf{dom}\,(X_L^{\smile};\Phi;X_R') & \text{commutativity} \\
 & = & \mathsf{dom}\,(X_L^{\smile};\Phi_L^{\smile};\Phi_R;X_R') & \\
 & \sqsubseteq & \mathsf{dom}\,(X_L^{\smile};\Phi_L^{\smile}) & \\
 & = & \mathsf{ran}\,(\Phi_L;X_L) &
\end{array}
$$

Now, $\mathsf{dom}\,\Psi' \sqcap \mathsf{ran}\,\mathrm{X}_\mathrm{L} \sqsubseteq \mathsf{ran}\,(\Phi_\mathrm{L};\mathrm{X}_\mathrm{L})$ is by definition of relative pseudo-complements equivalent to $\mathsf{dom}\,\Psi' \sqsubseteq \mathsf{ran}\,\mathrm{X}_\mathrm{L} \rightarrow \mathsf{ran}\,(\Phi_\mathrm{L};\mathrm{X}_\mathrm{L})$, so we have $\mathsf{dom}\,\Psi' \sqsubseteq \mathsf{ran}\,\Psi_\mathrm{L}$ by definition of the latter.

For factorisation, the following equalities are essential:

$$\begin{aligned}
\mathrm{X}_\mathrm{L}^{\smile};\Phi;\mathrm{X}_\mathrm{R}' &= \mathrm{X}_\mathrm{L}^{\smile};\mathrm{X}_\mathrm{L};\Psi' = \mathsf{ran}\,\mathrm{X}_\mathrm{L};\Psi' \\
\Phi^{\smile};\mathrm{X}_\mathrm{L};\Psi' &= \Phi^{\smile};\Phi;\mathrm{X}_\mathrm{R}' = \mathsf{ran}\,\Phi;\mathrm{X}_\mathrm{R}'
\end{aligned}$$

These imply:

$$\begin{aligned}
\mathrm{X}_\mathrm{R};Y &= \mathrm{X}_\mathrm{R};\mathrm{X}_\mathrm{R}^{\smile};\mathrm{X}_\mathrm{R}' \sqcup \mathrm{X}_\mathrm{R};\Psi^{\smile};\Psi' = (\Phi^{\smile};\mathrm{X}_\mathrm{L})^{\gg};\mathrm{X}_\mathrm{R}' \sqcup (\Phi^{\smile};\mathrm{X}_\mathrm{L})^{\boxtimes};\Psi' = \mathrm{X}_\mathrm{R}' \\
\Psi;Y &= \Psi;\mathrm{X}_\mathrm{R}^{\smile};\mathrm{X}_\mathrm{R}' \sqcup \Psi;\Psi^{\smile};\Psi' \\
&= (\mathrm{X}_\mathrm{L}^{\smile};\Phi)^{\boxtimes};\mathrm{X}_\mathrm{R}' \sqcup \Psi_\mathrm{L}^{\smile};\Psi_\mathrm{L};(\Phi^{\smile};\mathrm{X}_\mathrm{L})^{\ll};\Psi' = \mathsf{ran}\,\mathrm{X}_\mathrm{L};\Psi' \sqcup \mathsf{ran}\,\Psi_\mathrm{L};\Psi' = \Psi'
\end{aligned}$$

For the last equation we used $\mathsf{dom}\,\Psi' \sqsubseteq \mathsf{ran}\,\Psi_\mathrm{L}$, shown above. With factorisation, it is easy to show that $Y$ is univalent:

$$Y^{\smile};Y = \mathrm{X}_\mathrm{R}'^{\smile};\mathrm{X}_\mathrm{R};Y \sqcup \Psi'^{\smile};\Psi;Y \sqsubseteq \mathrm{X}_\mathrm{R}'^{\smile};\mathrm{X}_\mathrm{R}' \sqcup \Psi'^{\smile};\Psi' \sqsubseteq \mathbb{I}$$

Uniqueness: Assume a univalent relation $Y' : \mathcal{R} \leftrightarrow \mathcal{B}'$ with $\mathrm{X}_\mathrm{R};Y' = \mathrm{X}_\mathrm{R}'$ and $\Psi;Y' = \Psi'$. Then $Y' = Y$:

$$\begin{aligned}
Y' &= \mathsf{ran}\,\Psi_\mathrm{R};Y' \sqcup \mathsf{ran}\,\mathrm{X}_\mathrm{R};Y' = \Psi_\mathrm{R}^{\smile};\Psi_\mathrm{R};Y' \sqcup \mathrm{X}_\mathrm{R}^{\smile};\mathrm{X}_\mathrm{R};Y' \\
&= \Psi_\mathrm{R}^{\smile};\Psi_\mathrm{L};\Psi_\mathrm{L}^{\smile};\Psi_\mathrm{R};Y' \sqcup \mathrm{X}_\mathrm{R}^{\smile};\mathrm{X}_\mathrm{R}' = \Psi^{\smile};\Psi;Y' \sqcup \mathrm{X}_\mathrm{R}^{\smile};\mathrm{X}_\mathrm{R}' = \Psi^{\smile};\Psi' \sqcup \mathrm{X}_\mathrm{R}^{\smile};\mathrm{X}_\mathrm{R}' = Y \qquad \square
\end{aligned}$$

A general discussion of the single-pushout approach follows in Sect. 5.6.

## 5.5 Pullback Complements

Just like in the double-pushout approach, the left-hand-side square of a rewriting step in the double-pullback approach also poses the problem that the "wrong" arrows are to be constructed.

Assuming two mappings $\mathcal{D} \xrightarrow{P} \mathcal{B} \xrightarrow{R} \mathcal{A}$ to be given, we are therefore looking for a *pullback complement* $\mathcal{D} \xrightarrow{Q} \mathcal{C} \xrightarrow{S} \mathcal{A}$, that is, an object $\mathcal{C}$ and two mappings $Q : \mathcal{D} \to \mathcal{C}$ and $S : \mathcal{C} \to \mathcal{A}$ such that the resulting square is a pullback for $R$ and $S$.

$$\begin{tikzcd}
\mathcal{B} \arrow[r, "R"] & \mathcal{A} \\
\mathcal{D} \arrow[u, "P"] \arrow[r, "Q"] & \mathcal{C} \arrow[u, "S"']
\end{tikzcd}$$

The shape and names of this diagram will be used throughout the remainder of this section.

A necessary and sufficient condition for the existence of pullback complements in the category of concrete graphs has been given by Bauderon and Jacquet [BJ96, Jac99, BJ01]. This is an extremely complex condition formulated on the level of edges and nodes, postulating enumerations of pre-images satisfying certain compatibility conditions, and using quite intricate notation. Just to give an impression of the complexity of this componentwise approach to this problem, we cite the definition of Bauderon and Jacquet [BJ01, Def. 5] verbatim (without introducing the notation):



> A pair ($A \xrightarrow{a} B \xrightarrow{b} C$) of arrows is *coherent* if it has the three following properties:
>
> 1. for all $u \in V_C$, for all $i, j \in [1 \ldots \#[u]_b]$, $\#[u]^i_{ba} = \#[u]^j_{ba}$
> 2. for all $[u]^i_{ba}$, $i \in [1 \ldots \#[u]_b]$, there exists an enumeration,
>
>    $$[u]^i_{ba} = \{u^{i1}_{ba}, u^{i2}_{ba}, \ldots, u^{in}_{ba}\} \quad \text{for } n = \#[[u]^i_b]_a$$
>
>    such that for any edge $[u^{ii_1}_{ba}, v^{jj_1}_{ba}] \in E_A$ and any $i', j'$:
>
>    $$[u]^{i'}_{ba} Adj_A [v]^{j'}_{ba} \text{ implies } [u^{i'i_1}_{ba}, v^{j'j_1}_{ba}] \in E_A$$
>
> 3. given any two vertices $u, v \in V_C$, if one of the vertices $v^{ij}_{ba}$ is adjacent to $[u]^{i'}_{ba}$ then every edge of $B \mid ([v]_b \cup [u]_b)$ has a preimage under $a$.

In the relational approach, we can replace this with an abstract, component-free condition that is necessary and sufficient for the existence of pullback complements in the subcategory of mappings in arbitrary allegories. This condition is extremely simple, and thus offers valuable insight into the essence of pullback complements (we continue to use the name "coherent" proposed by Bauderon and Jacquet):

**Definition 5.5.1** [←118] Two mappings $\mathcal{D} \xrightarrow{P} \mathcal{B} \xrightarrow{R} \mathcal{A}$ are called *coherent* iff there exists an equivalence relation $\Theta : \mathcal{D} \leftrightarrow \mathcal{D}$ such that the following conditions hold:

(i) $P;P^{\smile} \sqcap \Theta \sqsubseteq \mathbb{I}$,

(ii) $\Theta;P = P;R;R^{\smile}$.

We also say then $P$ and $R$ are *coherent via* $\Theta$. $\square$

Note that (ii) implies $\Theta;P;P^{\smile} = P;R;R^{\smile};P^{\smile}$. Since with the right-hand side, also the left-hand side is an equivalence relation, and since the composition of two equivalence relations is always contained in their equivalence join, but not in any smaller equivalence relation, this means that $P;R;R^{\smile};P^{\smile}$ is the equivalence join of $\Theta$ and $P;P^{\smile}$. Together with (i) it then follows that $\Theta$ is a complement of $P;P^{\smile}$ in the lattice of all equivalence relations contained in $P;R;R^{\smile};P^{\smile}$. Since this lattice is, in general, not distributive, this complement need not be uniquely determined even if it exists. And not all complements in addition fulfil (ii). (For a counterexample, consider the equivalence relation joining just "1b" and "2b" in the drawing on page 119, where the relevant lattice of equivalence relations is not even modular.)

**Proposition 5.5.2** If $\mathcal{B} \xleftarrow{P} \mathcal{D} \xrightarrow{Q} \mathcal{C}$ is a pullback for $\mathcal{B} \xrightarrow{R} \mathcal{A} \xleftarrow{S} \mathcal{C}$, then $P$ and $R$ are coherent via $\Theta := Q;Q^{\smile}$.

**Proof**: With commutativity and alternative commutativity we obtain (ii):

$$P;R;R^{\smile} = Q;S;R^{\smile} = Q;Q^{\smile};P = \Theta;P$$

The condition (i) is part of the tabulation properties. □

If we know that $Q$ is surjective, then such an equivalence relation $\Theta$ already uniquely determines $Q$ (up to isomorphism) as a quotient projection for $\Theta$. In general, however, we only have the following:

**Proposition 5.5.3** [BJ96, Prop. 10] If $\mathcal{B} \xleftarrow{P} \mathcal{D} \xrightarrow{Q} \mathcal{C}$ is a pullback for $\mathcal{B} \xrightarrow{R} \mathcal{A} \xleftarrow{S} \mathcal{C}$ and $R$ is surjective, then $Q$ is surjective, too.

**Proof**: With a tabulation property, surjectivity of $R$, and totality of $S$:

$$Q^{\smile};Q = \mathsf{ran}\,(R;S^{\smile}) = \mathsf{ran}\,((\mathsf{ran}\,R);S^{\smile}) = \mathsf{ran}\,(S^{\smile}) = \mathsf{dom}\,S = \mathbb{I}$$ □

If $R$ is not surjective, then $Q$ need not be surjective, either, and the pullback complement object is not uniquely determined. However, for the pullback complement with surjective $Q$, the pullback complement object is a subobject of every other candidate. When looking for pullback complements, we shall therefore restrict our search to candidates with surjective $Q$.

With this restriction, surjectivity and univalence of $Q$ together with commutativity then also determine $S$:

$$S = Q^{\smile};Q;S = Q^{\smile};P;R \ .$$

Showing that all this then gives rise to a pullback complement is routine:

**Theorem 5.5.4** If there exists an equivalence relation $\Theta : \mathcal{D} \leftrightarrow \mathcal{D}$ such that $P$ and $R$ are coherent via $\Theta$, then a pullback complement $\mathcal{D} \xrightarrow{Q} \mathcal{C} \xrightarrow{S} \mathcal{A}$ for $\mathcal{D} \xrightarrow{P} \mathcal{B} \xrightarrow{R} \mathcal{A}$ is obtained as follows:

- Let $\mathcal{C}$ be a quotient of $\mathcal{D}$ for $\Theta$, with projection

$$Q : \mathcal{C} \leftrightarrow \mathcal{D} \qquad Q;Q^{\smile} = \Theta \qquad Q^{\smile};Q = \mathbb{I} \ .$$

- Define $S : \mathcal{C} \leftrightarrow \mathcal{A}$ as $S := Q^{\smile};P;R$.

**Proof**: $Q$ is total, univalent and surjective by construction. For $S$, univalence and totality are shown as follows:

$$S^{\smile};S \;=\; R^{\smile};P^{\smile};Q;Q^{\smile};P;R \;=\; R^{\smile};P^{\smile};P;R;R^{\smile};R \;=\; R^{\smile};P^{\smile};P;R \;\sqsubseteq\; \mathbb{I} \qquad \text{5.5.1.ii)}$$

$$S;S^{\smile} \;=\; Q^{\smile};P;R;R^{\smile};P^{\smile};Q \;=\; Q^{\smile};Q;Q^{\smile};P;P^{\smile};Q \;\sqsupseteq\; Q^{\smile};Q;Q^{\smile};Q \;=\; \mathbb{I} \qquad \text{5.5.1.ii)}, P \text{ total}$$

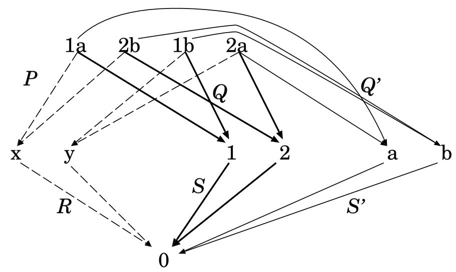
Commutativity and the tabulation conditions:

$$\begin{aligned}
Q;S &= Q;Q^{\smile};P;R = P;R;R^{\smile};R = P;R && \text{5.5.1.ii)}\\
S;R^{\smile} &= Q^{\smile};P;R;R^{\smile} = Q^{\smile};Q;Q^{\smile};P = Q^{\smile};P && \text{5.5.1.ii)}\\
\mathsf{ran}\,(S;R^{\smile}) &= \mathsf{ran}\,(Q^{\smile};P) = \mathsf{ran}\,P = P^{\smile};P && \text{alt. comm., } Q \text{ tot., } P \text{ univ.}\\
\mathsf{dom}\,(S;R^{\smile}) &= \mathsf{dom}\,(Q^{\smile};P) = \mathsf{dom}\,(Q^{\smile}) = \mathsf{ran}\,Q = Q^{\smile};Q && \text{alt. comm., } P \text{ tot., } Q \text{ univ.}\\
\mathbb{I} &= P;P^{\smile} \sqcap Q;Q^{\smile} && P,\ Q \text{ total; 5.5.1.i)}
\end{aligned}$$ □



In comparison with Bauderon and Jacquet's, ours is a much simpler formulation of the pullback complement condition. Although we do not attempt a proof of direct equivalence of the two conditions (too much notation would have to be introduced), they both share the property that they are in general not efficiently implementable — searching for an appropriate equivalence relation is as intractable as the equivalent search for a "coherent enumeration" of Bauderon and Jacquet.

Bauderon and Jacquet also show that a pullback complement for $\mathcal{D} \xrightarrow{P} \mathcal{B} \xrightarrow{R} \mathcal{A}$ with surjective $R$, if it exists, is unique up to isomorphism [BJ01, Prop. 4]. They do however not mention that this isomorphism need not be "natural", that is, it does not necessarily factorise the two pullback complements. This can be seen already with a simple direct product of discrete graphs:




Here, the targets of $Q$ and $Q'$ are clearly isomorphic, but no isomorphism $Y$ exists such that $Q' = Q;Y$.

So there are two reasons that make working with general pullback complements unsatisfactory: finding an equivalence relation determining the second projection $Q$ is, in general, computationally inefficient, and the resulting diagram is not even unique up to isomorphism.

However, if we consider how we want to use a pullback complement, then we notice that $R$ corresponds to the arrow mapping variable occurrences in a rule's left-hand side $L$ to variables in the gluing object $G$. If we restrict ourselves to left-linear rules, then there is only one occurrence in $L$ for every variable in $G$, and $R$ is injective. Fortunately, pullback complements for this special case are much simpler, as we are now going to see.

If $R$ is injective, then $Q$ is injective, too, so with the assumption that $Q$ is also surjective, we may set $Q = \mathbb{I}$ when constructing a pullback complement:

**Theorem 5.5.5** [←138] Given two mappings $\mathcal{D} \xrightarrow{P} \mathcal{B} \xrightarrow{R} \mathcal{A}$, if $R$ is injective, then a pullback complement $\mathcal{D} \xrightarrow{Q} \mathcal{C} \xrightarrow{S} \mathcal{A}$ is obtained as follows:

- Let $\mathcal{C} := \mathcal{D}$ and $Q := \mathbb{I}$.
- Define $S : \mathcal{C} \leftrightarrow \mathcal{A}$ as $S := P;R$.

**Proof**: $Q$ is a bijective mapping by definition. $S$ is a mapping by its definition as composition of two mappings. The tabulation properties follow easily with the properties of $P$ and $R$:

$$\begin{array}{rcll}
Q{;}S &=& P{;}R & \\
S{;}R\breve{} &=& P{;}R{;}R\breve{} \;=\; P \;=\; Q\breve{}{;}P & R \text{ total and inj.} \\
\mathsf{ran}\,(S{;}R\breve{}) &=& \mathsf{ran}\,(P{;}R{;}R\breve{}) \;=\; \mathsf{ran}\,P \;=\; P\breve{}{;}P & R \text{ tot., inj.; } P \text{ univ.} \\
\mathsf{dom}\,(S{;}R\breve{}) &=& \mathsf{dom}\,(P{;}R{;}R\breve{}) \;=\; \mathsf{dom}\,P \;=\; \mathbb{I} \;=\; Q\breve{}{;}Q & R,\ P \text{ total} \\
P{;}P\breve{} \sqcap Q{;}Q\breve{} &=& P{;}P\breve{} \sqcap \mathbb{I} \;=\; \mathbb{I} & P \text{ total} \qquad \square
\end{array}$$

In contrast to the general case, this trivial pullback complement is also more "well-behaved" in that it factorises other candidates:

**Proposition 5.5.6** If $\mathcal{D}\xrightarrow{Q}\mathcal{C}\xrightarrow{S}\mathcal{A}$ and $\mathcal{D}\xrightarrow{Q'}\mathcal{C}'\xrightarrow{S'}\mathcal{A}$ are two pullback complements for $\mathcal{D}\xrightarrow{P}\mathcal{B}\xrightarrow{R}\mathcal{A}$, and if $Q$ is injective and surjective, then

$$Y := Q\breve{}{;}Q'$$

is a mapping and $Q' = Q{;}Y$ and $S = Y{;}S'$.

**Proof**: Injectivity of $Q$ transfers univalence from $Q'$ to $Y$:

$$Y\breve{}{;}Y = Q'\breve{}{;}Q{;}Q\breve{}{;}Q' = Q'\breve{}{;}Q' \sqsubseteq \mathbb{I}$$

Surjectivity of $Q$ transfers totality from $Q'$ to $Y$:

$$Y{;}Y\breve{} = Q\breve{}{;}Q'{;}Q'\breve{}{;}Q \sqsupseteq Q\breve{}{;}Q = \mathbb{I}$$

For factorisation of $S$, we additionally need commutativity:

$$\begin{array}{rcl}
Q{;}Y &=& Q{;}Q\breve{}{;}Q' \;=\; Q' \\
Y{;}S' &=& Q\breve{}{;}Q'{;}S' \;=\; Q\breve{}{;}P{;}R \;=\; Q\breve{}{;}Q{;}S \;=\; S
\end{array} \qquad \square$$

## 5.6 Pushouts of Partial Functions

As mentioned in the introduction, the single-pushout approach of [Löw90, Löw93, EL93] employs pushouts in categories of *partial graph homomorphisms* for its derivation steps.

If $\mathcal{A}$ is an object in a Dedekind category $\mathbf{D}$ such that for every partial identity $u : \mathsf{PId}\,\mathcal{A}$ a subobject exists, then partial morphisms starting from $\mathcal{A}$, where partial morphisms are taken with respect to the category $\mathsf{Map}\,\mathbf{D}$ of mappings, are equivalent (up to isomorphism of subobjects) to univalent relations starting from $\mathcal{A}$ in $\mathbf{D}$.

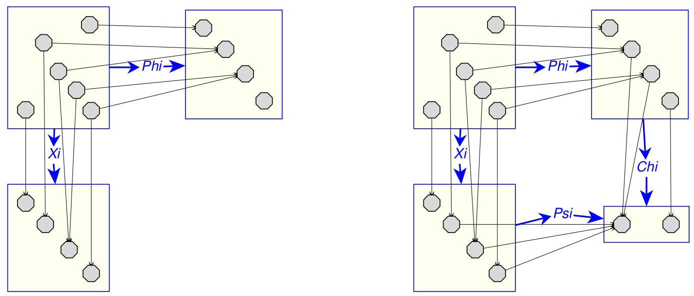
Therefore, we are now considering the category $\mathsf{Pfn}\,\mathbf{D}$ of *partial functions in* $\mathbf{D}$, which is the subcategory of $\mathbf{D}$ with all objects of $\mathbf{D}$, and as arrows only univalent relations.

With an eye to the application of the pushout in $\mathsf{Pfn}\,\mathbf{D}$ as rewriting step, we shall use the following names throughout this section:

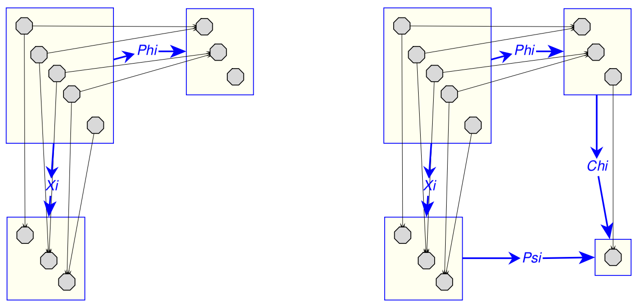


$$\begin{tikzcd}
\mathcal{L} \arrow[r, "\Phi"] \arrow[d, "\Xi"'] & \mathcal{R} \arrow[d, "\mathrm{X}"] \arrow[ddr, "\mathrm{X}'"] & \\
\mathcal{A} \arrow[r, "\Psi"] \arrow[drr, "\Psi'"'] & \mathcal{B} \arrow[dr, dotted, "\mathrm{Y}"] & \\
& & \mathcal{B}'
\end{tikzcd}$$

For illustration, we show a span $\mathcal{R}\xleftarrow{\Phi}\mathcal{L}\xrightarrow{\Xi}\mathcal{A}$ of two non-total morphisms in the category $\mathsf{Pfn}\,(\mathsf{sigGraph\text{-}GS}_{Rel})$ of partial graph homomorphisms, and its pushout — not yet involving any complicated effects.




Now we show a variant where we add to the central group of the $\mathcal{L}$ above a single node with no image via $\Phi$, and we let it share its image via $\Xi$ with one of the other nodes. This partiality then propagates to the whole group:




In view of such subtle effects, an important question is whether the pushout construction in categories of partial morphisms can be reduced to simpler constructions.

For relating the pushout in categories of partial $\Sigma$-algebra morphisms to simpler concepts, Löwe originally proposed the following construction [Löw90, LE91, Löw93] (slightly adapted to our formalism):

- Assume a span $\mathcal{R} \xleftarrow{\Phi} \mathcal{L} \xrightarrow{\Xi} \mathcal{A}$ of *partial* morphisms.
- Construct the “gluing object” $\mathcal{L}'$ as a subobject of $\mathcal{L}$ with injection $\lambda$ such that
  - $\mathsf{ran}\,\lambda \sqsubseteq \mathsf{dom}\,\Phi \sqcap \mathsf{dom}\,\Xi$, and
  - for all items $x$ of $\mathcal{L}'$ and $y$ of $\mathcal{L}$, if $\Phi$ or $\Xi$ identifies $\lambda(x)$ with $y$, then $y$ is in $\mathsf{ran}\,\lambda$, too.
- Construct the “scopes” $\mathcal{R}'$ and $\mathcal{A}'$ via
  - $\iota : \mathcal{R}' \rightarrowtail \mathcal{R}$ as subobject injection for $\mathsf{ran}\,\Phi \to \mathsf{ran}\,(\lambda;\Phi)$, and
  - $\kappa : \mathcal{A}' \rightarrowtail \mathcal{A}$ as subobject injection for $\mathsf{ran}\,\Xi \to \mathsf{ran}\,(\lambda;\Xi)$.

  $\Phi' := \lambda;\Phi;\iota^{\smile}$ and $\Xi' := \lambda;\Xi;\kappa^{\smile}$ are then the resulting *total* restrictions of $\Phi$ and $\Xi$.
- Let $\mathcal{R}' \xrightarrow{X_0} \mathcal{B} \xleftarrow{\Psi_0} \mathcal{A}'$ be a pushout in the category of *total* morphisms for the span $\mathcal{R}' \xleftarrow{\Phi'} \mathcal{L}' \xrightarrow{\Xi'} \mathcal{A}'$. (This is a gluing for $\Phi'^{\smile};\Xi'$.)
- Define $X := \iota^{\smile};X_0$ and $\Psi := \kappa^{\smile};\Psi_0$.
- Then $\mathcal{R} \xrightarrow{X} \mathcal{B} \xleftarrow{\Psi} \mathcal{A}$ is a pushout for $\mathcal{R} \xleftarrow{\Phi} \mathcal{L} \xrightarrow{\Xi} \mathcal{A}$ in the category of *partial* morphisms.

$$\begin{tikzcd}
\mathcal{L} \arrow[rr, "\Phi"] \arrow[dd, "\Xi"'] & & \mathcal{R} \arrow[dd, bend left=60, "X"] \\
 & \mathcal{L}' \arrow[ul, "\lambda"'] \arrow[r, "\Phi'"] \arrow[d, "\Xi'"'] & \mathcal{R}' \arrow[u, "\iota"'] \arrow[d, "X_0"] \\
\mathcal{A} \arrow[rr, bend right=40, "\Psi"'] & \mathcal{A}' \arrow[l, "\kappa"'] \arrow[r, "\Psi_0"] & \mathcal{B}
\end{tikzcd}$$

In our reformulated definition of the scopes, we obtained an obvious analogy to the pushout complement. However, the construction of the gluing object is still on the component level. Achieving a component-free definition of the gluing object requires a solution via different tools than those used for the scopes.

In fact, it turns out that it is not even necessary to consider a gluing object, as long as the relevant restrictions are imposed.

A purely category-theoretic formulation can be found in the graph transformation handbook chapter about the single-pushout approach [EHK+97]. Here, pushouts in categories of partial morphisms are constructed using two pushouts in the corresponding category of total morphisms and one co-equaliser in the category of partial morphisms:

- Assume a span $\mathcal{R} \xleftarrow{\Phi} \mathcal{L} \xrightarrow{\Xi} \mathcal{A}$ of *partial* morphisms, factored via appropriate subobjects as follows: $\Xi = \iota^{\smile};\Xi_0$ and $\Phi = \kappa^{\smile};\Phi_0$.



- Let $\mathcal{L} \xrightarrow{\Xi_1} \mathcal{A}' \xleftarrow{\upsilon} \mathcal{A}$ be a pushout of *total* morphisms for $\mathcal{L} \xleftarrow{\iota} \mathsf{dom}\,\Xi \xrightarrow{\Xi_0} \mathcal{A}$. (This is just a gluing for $\Xi$.)
- Let $\mathcal{R} \xrightarrow{X_0} \mathcal{B}_0 \xleftarrow{\Psi_0} \mathcal{A}'$ be a pushout of *total* morphisms for $\mathcal{R} \xleftarrow{\Phi_0} \mathsf{dom}\,\Phi \xrightarrow{\kappa;\Xi_1} \mathcal{A}'$. (This is a gluing for $\Phi^{\smile};\Xi_1$.)
- Let $E : \mathcal{B}_0 \twoheadrightarrow \mathcal{B}$ be a co-equaliser in the category of *partial* morphisms for $\Phi;X_0$ and $\Xi_1;\Psi_0$.
- Let $X := X_0;E$ and $\Psi := \upsilon;\Psi_0;E$.
- Then $\mathcal{R} \xrightarrow{X} \mathcal{B} \xleftarrow{\Psi} \mathcal{A}$ is a pushout for $\mathcal{R} \xleftarrow{\Phi} \mathcal{L} \xrightarrow{\Xi} \mathcal{A}$ in the category of *partial* morphisms.

$$\begin{tikzcd}
\mathsf{dom}\,\Xi \arrow[r, "\iota"] \arrow[d, "\Xi_0"'] & \mathcal{L} \arrow[rr, bend left, "\Phi"] \arrow[dl, "\Xi"'] \arrow[d, "\Xi_1"] & \mathsf{dom}\,\Phi \arrow[l, "\kappa"'] \arrow[r, "\Phi_0"] & \mathcal{R} \arrow[d, "X_0"] \arrow[ddr, "X"] & \\
\mathcal{A} \arrow[r, "\upsilon"] \arrow[ddrrrr, phantom] \arrow[drrrr, bend right, "\Psi"'] & \mathcal{A}' \arrow[rr, "\Psi_0"] & & \mathcal{B}_0 \arrow[dr, dotted, "E"] & \\
 & & & & \mathcal{B}
\end{tikzcd}$$

Although it may not be immediately obvious, it is the co-equaliser that now takes on the rôle of domain restriction that is played by the gluing object in Löwe’s original construction. (It is, however, quite unsatisfactory that this co-equaliser still has to be taken in the category of *partial* morphisms.)

This co-equaliser implements a restriction that is very close to that implemented by a *symmetric quotient*. Symmetric quotients have originally been introduced by Gunther Schmidt and his group in the context of heterogeneous relation algebras, and were then used to formalise power objects in relation algebras and other domain constructions useful for programming language semantics, see [BSZ86, BSZ89, Zie91]. In this context, the definition is usually given using the complement operation:

$$\mathrm{syq}(Q, S) := \overline{Q^{\smile};\overline{S}} \sqcap \overline{\overline{Q^{\smile}};S}$$

However, symmetric quotients can already be defined in arbitrary allegories, albeit they do not necessarily exist for all arguments:

**Definition 5.6.1** In an allegory, the *symmetric quotient* $\mathrm{syq}(Q, S) : B \leftrightarrow C$ of two relations $Q : A \leftrightarrow B$ and $S : A \leftrightarrow C$ is defined by

$$X \sqsubseteq \mathrm{syq}(Q, S) \iff Q;X \sqsubseteq S \;\text{ and }\; X;S^{\smile} \sqsubseteq Q^{\smile} \qquad \text{for all } X : B \leftrightarrow C\ . \qquad \square$$

In Dedekind categories, symmetric quotients are always defined and the following hold:

$$\begin{aligned}\mathrm{syq}(Q,S) &= Q\backslash S \sqcap Q^{\smile}/S^{\smile} = Q\backslash S \sqcap (S\backslash Q)^{\smile}\\ &= \bigsqcup\{X \mid Q;X \sqsubseteq S \text{ and } X;S^{\smile} \sqsubseteq Q^{\smile}\}\ .\end{aligned}$$

Modulo conversion of the arguments, the symmetric quotient is exactly the *symmetric division* as defined by Freyd and Scedrov for division allegories [FS90, 2.35]. (Riguet had introduced the unary operation of "noyeau", which can now be seen as defined by $\mathsf{noy}(R) = \mathrm{syq}(R,R)$, in [Rig48].)

Used for concrete relations, the symmetric quotient relates elements from the ranges of the two relations $R$ and $S$ iff they have the same inverse image under $R$ respectively $S$:

$$(r,s) \in \mathrm{syq}(R,S) \qquad \Longleftrightarrow \qquad \forall x : (x,r) \in R \leftrightarrow (x,s) \in S$$

For a few properties of symmetric quotients relevant in the remainder of this section see Sect. A.5; more information can also be found in [SS93, Sect. 4.4] and in [FK98].

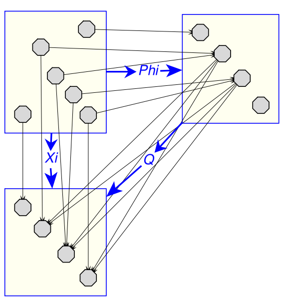
For application in the single-pushout construction, we shall need a special variant of symmetric quotients:

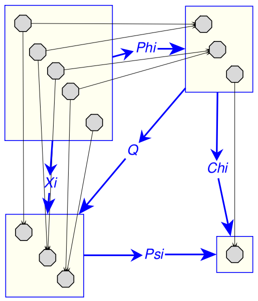
**Definition 5.6.2** Let two morphisms $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given.

Let $W := \Phi^{\smile};\Xi$. The *saturated symmetric quotient* $\mathsf{satSyq}(\Phi,\Xi) : \mathcal{R} \leftrightarrow \mathcal{A}$ of $\Phi$ and $\Xi$ is defined by the following:

$$\mathsf{satSyq}(\Phi,\Xi) \quad := \quad \mathsf{ran}\,\Phi;\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd});\mathsf{ran}\,\Xi\ . \qquad \square$$

Put into the context of the single-pushout approach, the saturated symmetric quotient relates every node in $\mathsf{ran}\,\Phi$ with all those nodes in $\mathsf{ran}\,\Xi$ that will share the image with this node. In the following drawing, we show the saturated symmetric quotient $Q := \mathsf{satSyq}(\Phi,\Xi)$ for the two examples from above:




We have seen in the introduction that in the single-pushout approach, deletion takes priority over preservation. In the second example above, this has relatively far-reaching consequences and is reflected by the fact that $Q := \mathsf{satSyq}(\Phi,\Xi) = \perp\!\!\!\perp$, so only the node outside the image of $\mathsf{ran}\,\Phi$ is copied into the result of the single-pushout step.



Using the fact that symmetric quotients are always difunctional, we easily see that saturated symmetric quotients are difunctional, too — letting $Q := \mathsf{satSyq}(\Phi,\Xi)$ and $Q_0 := \mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd})$ we have that $Q_0$ is difunctional, and then we obtain:

$$\begin{aligned}Q;Q^{\smile};Q &= \mathsf{ran}\,\Phi;Q_0;\mathsf{ran}\,\Xi;Q_0^{\smile};\mathsf{ran}\,\Phi;Q_0;\mathsf{ran}\,\Xi\\ &\sqsubseteq \mathsf{ran}\,\Phi;Q_0;Q_0^{\smile};Q_0;\mathsf{ran}\,\Xi = \mathsf{ran}\,\Phi;Q_0;\mathsf{ran}\,\Xi = Q\end{aligned}$$

Before we go on to show more properties of saturated symmetric quotients, we first show some useful facts that hold in the single-pushout setup:

**Lemma 5.6.3** [←126, 128] Let two univalent relations $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given, and let $W := \Phi^{\smile};\Xi$.

If for a cospan $\mathcal{R} \xrightarrow{\mathrm{X}} \mathcal{B} \xleftarrow{\Psi} \mathcal{L}$ of univalent relations, commutativity $\Phi;\mathrm{X} = \Xi;\Psi$ holds, then the following hold, too:

(i) $W;\Psi = \mathsf{ran}\,\Phi;\mathrm{X}$ and $W^{\smile};\mathrm{X} = \mathsf{ran}\,\Xi;\Psi$.

(ii) $W^{\rhd};\mathrm{X} = \mathrm{X}$ , $\quad W^{\lhd};\Psi = \Psi$ , $\quad W^{\boxast};\Psi = \mathsf{ran}\,\Phi;\mathrm{X}$.

(iii) $\Phi;\mathsf{dom}\,\mathrm{X} \sqsubseteq \Xi;\Xi^{\smile};\Phi$ and $\Xi;\mathsf{dom}\,\Psi \sqsubseteq \Phi;\Phi^{\smile};\Xi$.

(iv) $\Phi;W^{\rhd}$ and $\Xi;W^{\lhd}$ are difunctional.

**Proof**:

(i) $W;\Psi = \Phi^{\smile};\Xi;\Psi = \Phi^{\smile};\Phi;\mathrm{X} = \mathsf{ran}\,\Phi;\mathrm{X}$ and $W^{\smile};\mathrm{X} = \Xi^{\smile};\Phi;\mathrm{X} = \Xi^{\smile};\Xi;\Psi = \mathsf{ran}\,\Xi;\Psi$

(ii) follows immediately from (i).

(iii) With univalence of $\Phi$, commutativity, and a modal rule we obtain:

$$\Phi;\mathsf{dom}\,\mathrm{X} = \Phi;(\mathbb{I} \sqcap \mathrm{X};\mathrm{X}^{\smile}) = \Phi \sqcap \Phi;\mathrm{X};\mathrm{X}^{\smile} = \Phi \sqcap \Xi;\Psi;\mathrm{X}^{\smile} \sqsubseteq \Xi;\Xi^{\smile};\Phi$$

(iv) With univalence of $\Phi$, we obtain:

$$\Phi;W^{\rhd};W^{\rhd};\Phi^{\smile};\Phi;W^{\rhd} \sqsubseteq \Phi;W^{\rhd};W^{\rhd};W^{\rhd} = \Phi;W^{\rhd}$$

Difunctionality of $\Xi;W^{\lhd}$ is shown in the same way. $\square$

**Lemma 5.6.4** [←126, 127, 129] Let two morphisms $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given, and define $W := \Phi^{\smile};\Xi$. Then the following holds for the saturated symmetric quotient $Q := \mathsf{satSyq}(\Phi,\Xi)$:

$$\mathsf{dom}\,Q;W^{\boxast} = Q = W^{\boxast};\mathsf{ran}\,Q$$

**Proof**:

$$\begin{aligned}Q &= \mathsf{ran}\,\Phi;\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd});\mathsf{ran}\,\Xi\\ &= \mathsf{ran}\,(\Phi;W^{\rhd});\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd});\mathsf{ran}\,\Xi && \mathsf{ran}\,\Phi = \mathsf{ran}\,(\Phi;W^{\rhd})\\ &= W^{\rhd};\Phi^{\smile};\Phi;W^{\rhd};\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd});\mathsf{ran}\,\Xi && \Phi;W^{\rhd}\text{ difctl., A.5.2}\\ &= W^{\rhd};\Phi^{\smile};\Xi;W^{\lhd};\mathsf{ran}\,(\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd}));\mathsf{ran}\,\Xi && \text{A.5.1.ii)}\\ &= W^{\boxast};\mathsf{ran}\,(\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd});\mathsf{ran}\,\Xi)\\ &= W^{\boxast};\mathsf{ran}\,(\mathsf{ran}\,\Phi;\mathrm{syq}(\Phi;W^{\rhd},\Xi;W^{\lhd});\mathsf{ran}\,\Xi) && \text{A.5.1.iii)}\\ &= W^{\boxast};\mathsf{ran}\,Q\end{aligned}$$

The other equation is shown in the same way. $\square$

**Lemma 5.6.5** [←127] Let two univalent relations $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given, and let $Q := \mathsf{satSyq}(\Phi, \Xi)$.

If for a cospan $\mathcal{R} \xrightarrow{X} \mathcal{B} \xleftarrow{\Psi} \mathcal{L}$ of univalent relations, commutativity $\Phi;X = \Xi;\Psi$ holds, then $Q;\Psi = \mathsf{dom}\,Q;X$ and $Q^{\smile};X = \mathsf{ran}\,Q;\Psi$ hold, too.

**Proof**:
$$\begin{array}{rcll} Q;\Psi &=& \mathsf{dom}\,Q;W^{\boxast};\Psi & 5.6.4\\ &=& \mathsf{dom}\,Q;\mathsf{ran}\,\Phi;X & \text{Lemma 5.6.3.i)}\\ &=& \mathsf{dom}\,Q;X & \mathsf{dom}\,Q \sqsubseteq \mathsf{ran}\,\Phi \end{array}$$

The equation $Q^{\smile};X = \mathsf{ran}\,Q;\Psi$ follows in the same way. □

We now propose a construction that preserves part of the structure of Löwe's original construction, but for which we are able to give a component-free definition that replaces Löwe's "gluing object" in its rôle as determining the "scopes" with the saturated symmetric quotient:

**Definition 5.6.6** Let two univalent relations $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given, and let $Q := \mathsf{satSyq}(\Phi, \Xi)$.

Let $\mathcal{R}'$ be the subobject for $\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q$, with subobject injection $\iota : \mathcal{R}' \to \mathcal{R}$, and let $\mathcal{A}'$ be the subobject for $\mathsf{ran}\,\Xi \to \mathsf{ran}\,Q$, with subobject injection $\kappa : \mathcal{A}' \to \mathcal{A}$.

Then let $\mathcal{R}' \xrightarrow{X_0} \mathcal{B} \xleftarrow{\Psi_0} \mathcal{A}'$ be a gluing for $\iota^{\smile};Q;\kappa$, and define

$$X := \iota^{\smile};X_0 \qquad \text{and} \qquad \Psi := \kappa^{\smile};\Psi_0 \ .$$

Then $\mathcal{R} \xrightarrow{X} \mathcal{B} \xleftarrow{\Psi} \mathcal{A}$ is called the *reduced gluing* for $\Phi$ and $\Xi$. □

$$\begin{tikzcd}
\mathcal{L} \arrow[rr, "\Phi"] \arrow[dd, "\Xi"'] & & \mathcal{R} \arrow[ddll, "Q"'] \arrow[dd, bend left=60, "X"] \\
& & \mathcal{R}' \arrow[u, "\iota"'] \arrow[d, "X_0"] \arrow[dl, "\iota^{\smile};Q;\kappa"'] \\
\mathcal{A} \arrow[r, bend right=40, "\Psi"'] & \mathcal{A}' \arrow[l, "\kappa"'] \arrow[r, "\Psi_0"] & \mathcal{B}
\end{tikzcd}$$

Obviously, the detailed behaviour of $\Phi$ and $\Xi$ enters this definition only via their ranges and saturated symmetric quotient, so the corresponding algebraic characterisation may start from there — gathering the range information into $U := \Phi^{\smile};\mathbb{T};\Xi$.



**Definition 5.6.7** [←128] Let two difunctional relations $Q, U : \mathcal{R} \leftrightarrow \mathcal{A}$ with $Q \sqsubseteq U$ be given.

A cospan $\mathcal{R} \xrightarrow{X} \mathcal{B} \xleftarrow{\Psi} \mathcal{A}$ is called a *restricted gluing for $Q$ in $U$* if:

$$\begin{array}{rcl} X;\Psi^{\smile} &=& Q\\ X;X^{\smile} &=& (\mathsf{dom}\,U \to \mathsf{dom}\,Q) \sqcup Q;Q^{\smile}\\ \Psi;\Psi^{\smile} &=& (\mathsf{ran}\,U \to \mathsf{ran}\,Q) \sqcup Q^{\smile};Q\\ X^{\smile};X \sqcup \Psi^{\smile};\Psi &=& \mathbb{I} \end{array}$$ □

Since $Q$ is difunctional we have $\mathbb{I} \sqcup Q;Q^{\smile} = Q^{\triangleright}$, which helps to show how close this is to the gluing definition of Def. 5.3.2.

**Lemma 5.6.8** Let two univalent relations $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given, and let $W := \Phi^{\smile};\Xi$ and $Q := \mathsf{satSyq}(\Phi, \Xi)$.

A restricted gluing $\mathcal{R} \xrightarrow{X} \mathcal{B} \xleftarrow{\Psi} \mathcal{A}$ for $Q$ in $\Phi^{\smile};\mathbb{T};\Xi$ is then a pushout of partial functions.

**Proof**: Assume that for a cospan $\mathcal{R} \xrightarrow{X'} \mathcal{B} \xleftarrow{\Psi'} \mathcal{L}$ of univalent relations, commutativity $\Phi;X' = \Xi;\Psi'$ holds. Then we have to show that there exists a unique univalent relation $Y : \mathcal{B} \leftrightarrow \mathcal{B}'$ such that $X;Y = X'$ and $\Psi;Y = \Psi'$.

Define:
$$Y := \Psi^{\smile};\Psi' \sqcup X^{\smile};X'$$

Then Lemma 5.6.5 together with $S \sqsubseteq R \to S$ yields:

$$\begin{array}{rcl} X;Y &=& X;\Psi^{\smile};\Psi' \sqcup X;X^{\smile};X'\\ &=& Q;\Psi' \sqcup ((\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q) \sqcup Q;Q^{\smile});X'\\ &=& Q;\Psi' \sqcup (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q);X' \sqcup Q;Q^{\smile};X'\\ &=& Q;\Psi' \sqcup (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q);X' \sqcup Q;\mathsf{ran}\,Q;\Psi'\\ &=& Q;\Psi' \sqcup (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q);X'\\ &=& \mathsf{dom}\,Q;X' \sqcup (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q);X' \ = \ (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q);X' \ ,\\ \Psi;Y &=& \Psi;\Psi^{\smile};\Psi' \sqcup \Psi;X^{\smile};X'\\ &=& ((\mathsf{ran}\,\Xi \to \mathsf{ran}\,Q) \sqcup Q^{\smile};Q);\Psi' \sqcup Q^{\smile};X' \ = \ (\mathsf{ran}\,\Xi \to \mathsf{ran}\,Q);\Psi' \ . \end{array}$$

For showing the factorisation equalities we therefore have to show the inclusions $\mathsf{dom}\,X' \sqsubseteq (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q)$ and $\mathsf{dom}\,\Psi' \sqsubseteq (\mathsf{ran}\,\Xi \to \mathsf{ran}\,Q)$. We only show the first:

$$\begin{array}{cll} & \mathsf{dom}\,X' \sqsubseteq (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q) & \\ \Leftrightarrow & \mathsf{ran}\,\Phi \sqcap \mathsf{dom}\,X' \sqsubseteq \mathsf{dom}\,Q & \\ \Leftrightarrow & \mathsf{dom}\,(\Phi^{\smile};\Phi;X') \sqsubseteq \mathsf{dom}\,Q & \\ \Leftrightarrow & \mathsf{dom}\,(\Phi^{\smile};\Phi;X');W^{\boxast} \sqsubseteq \mathsf{dom}\,Q;W^{\boxast} & \mathsf{dom}\,(\Phi^{\smile};\Phi;X') \sqsubseteq \mathsf{dom}\,W\\ \Leftrightarrow & \mathsf{dom}\,(\Phi^{\smile};\Phi;X');W^{\boxast} \sqsubseteq Q & \text{Lemma 5.6.4}\\ \Leftrightarrow & \mathsf{dom}\,(\Phi^{\smile};\Phi;X');W^{\boxast} \sqsubseteq \mathrm{syq}(\Phi;W^{\triangleright}, \Xi;W^{\triangleleft}) & \mathsf{ran}\,W \sqsubseteq \mathsf{ran}\,\Xi \end{array}$$

By the definition of symmetric quotients, this is equivalent to the conjunction of the following two inclusions:

$$\Phi;W^{\rhd};\mathsf{dom}\,(\Phi^{\smile};\Phi;\mathrm{X}');W^{\boxast} \sqsubseteq \Xi;W^{\lhd} \quad\text{and}\quad \mathsf{dom}\,(\Phi^{\smile};\Phi;\mathrm{X}');W^{\boxast};W^{\lhd};\Xi^{\smile} \sqsubseteq W^{\rhd};\Phi^{\smile}\ ,$$

which we now show separately:

$$\begin{array}{rcll}
\Phi;W^{\rhd};\mathsf{dom}\,(\mathrm{X}');W^{\boxast} & = & \Phi;\mathsf{dom}\,(\mathrm{X}');W^{\rhd};\mathsf{dom}\,(\mathrm{X}');W^{\boxast} & \text{5.6.3.ii)} \\
& \sqsubseteq & \Xi;\Xi^{\smile};\Phi;W^{\boxast} & \text{Lemma 5.6.3.iii)} \\
& = & \Xi;W^{\lhd} & \\
\mathsf{dom}\,(\Phi^{\smile};\Phi;\mathrm{X}');W^{\boxast};W^{\lhd};\Xi^{\smile} & = & \mathsf{dom}\,(\mathrm{X}');W^{\boxast};\Xi^{\smile} & \\
& = & W^{\boxast};\mathsf{dom}\,(\Psi');\Xi^{\smile} & \text{5.6.3.ii)} \\
& \sqsubseteq & W^{\boxast};\Xi^{\smile};\Phi;\Phi^{\smile} & \text{Lemma 5.6.3.iii)} \\
& = & W^{\rhd};\Phi^{\smile} &
\end{array}$$

This proves $\mathsf{dom}\,\mathrm{X}' \sqsubseteq (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q)$, and therewith $\mathrm{X};\mathrm{Y} = \mathrm{X}'$. Factorisation of $\Psi'$ is obtained analogously.

With the factorisation equations, univalence of Y is straightforward:

$$\mathrm{Y}^{\smile};\mathrm{Y} = (\Psi'^{\smile};\Psi \sqcup \mathrm{X}'^{\smile};\mathrm{X});\mathrm{Y} = \Psi'^{\smile};\Psi;\mathrm{Y} \sqcup \mathrm{X}'^{\smile};\mathrm{X};\mathrm{Y} = \Psi'^{\smile};\Psi' \sqcup \mathrm{X}'^{\smile};\mathrm{X}' \sqsubseteq \mathbb{I}$$

Now assume the existence of another relation $\mathrm{Y}' : \mathcal{B} \nrightarrow \mathcal{B}'$ such that $\mathrm{X};\mathrm{Y}' = \mathrm{X}'$ and $\Psi;\mathrm{Y}' = \Psi'$. Then these equations, together with the last equation of Def. 5.6.7, imply:

$$\mathrm{Y} = \Psi^{\smile};\Psi' \sqcup \mathrm{X}^{\smile};\mathrm{X}' = \Psi^{\smile};\Psi;\mathrm{Y}' \sqcup \mathrm{X}^{\smile};\mathrm{X};\mathrm{Y}' = (\Psi^{\smile};\Psi \sqcup \mathrm{X}^{\smile};\mathrm{X});\mathrm{Y}' = \mathrm{Y}' \qquad \square$$

The pushout property also implies that our definition of restricted gluings is monomorphic.

Finally, we can prove that the reduced gluing construction, which we think is simpler than the categorical construction using a co-equaliser of partial morphisms, and also than the (component-wise) gluing-object construction of Löwe, indeed produces a pushout of partial functions:

**Theorem 5.6.9** Let two univalent relations $\Phi : \mathcal{L} \to \mathcal{R}$ and $\Xi : \mathcal{L} \to \mathcal{A}$ in a strict Dedekind category be given, and let $W := \Phi^{\smile};\Xi$ and $Q := \mathsf{satSyq}(\Phi, \Xi)$.

A reduced gluing $\mathcal{R} \xrightarrow{\mathrm{X}} \mathcal{B} \xleftarrow{\Psi} \mathcal{A}$ for $\mathcal{R} \xleftarrow{\Phi} \mathcal{L} \xrightarrow{\Xi} \mathcal{A}$ then commutes and is a restricted gluing for $Q$ in $\Phi^{\smile};\mathbb{T};\Xi$.

**Proof**: X and $\Psi$ are by definition partial functions. Since $Q$ is difunctional, $\iota;Q;\kappa^{\smile}$ is difunctional, too, because of univalence of $\iota$ and $\kappa$:

$$\iota;Q;\kappa^{\smile};\kappa;Q^{\smile};\iota^{\smile};\iota;Q;\kappa^{\smile} \sqsubseteq \iota;Q;Q^{\smile};Q;\kappa^{\smile} = \iota;Q;\kappa^{\smile}$$

With 2.6.9, we obtain:

$$\mathsf{dom}\,Q \sqsubseteq \mathsf{ran}\,\iota \qquad\text{and}\qquad \mathsf{ran}\,Q \sqsubseteq \mathsf{ran}\,\kappa \qquad (*)$$



For commutativity, we first consider one inclusion:

$$\begin{array}{rcll}
\Xi;\Psi \sqsubseteq \Phi;\mathrm{X} & \Leftrightarrow & \Xi;\kappa^{\smile};\Psi_0 \sqsubseteq \Phi;\iota^{\smile};\mathrm{X}_0 & \\
& \Leftrightarrow & \Phi^{\smile};\Xi;\kappa^{\smile} \sqsubseteq \iota^{\smile};\mathrm{X}_0;\Psi_0^{\smile} & \text{Lemma A.1.2.iii)} \\
& \Leftrightarrow & W;\kappa^{\smile} \sqsubseteq \iota^{\smile};(\iota;Q;\kappa^{\smile})^{\boxast} & \\
& \Leftrightarrow & W;\kappa^{\smile} \sqsubseteq \iota^{\smile};\iota;Q;\kappa^{\smile} & \\
& \Leftrightarrow & W;\kappa^{\smile};\kappa \sqsubseteq \iota^{\smile};\iota;Q & \text{Lemma A.1.2.iii)} \\
& \Leftrightarrow & W;\mathsf{ran}\,Q \sqsubseteq Q & (*)
\end{array}$$

This last inclusion holds because of Lemma 5.6.4. The opposite inclusion is shown in the same way, so we have commutativity $\Xi;\Psi = \Phi;\mathrm{X}$.

The gluing properties translate into the following, using $(*)$:

$$\begin{array}{rcl}
\mathrm{X};\Psi^{\smile} & = & \iota^{\smile};\mathrm{X}_0;\Psi_0^{\smile};\kappa \;=\; \iota^{\smile};(\iota;Q;\kappa^{\smile})^{\boxast};\kappa \;=\; \iota^{\smile};\iota;Q;\kappa^{\smile};\kappa \;=\; Q \\
\mathrm{X};\mathrm{X}^{\smile} & = & \iota^{\smile};\mathrm{X}_0;\mathrm{X}_0^{\smile};\iota \;=\; \iota^{\smile};(\mathbb{I} \sqcup (\iota;Q;\kappa^{\smile})^{\boxast};\kappa;Q^{\smile};\iota^{\smile});\iota \;=\; \iota^{\smile};(\mathbb{I} \sqcup \iota;Q;\kappa^{\smile};\kappa;Q^{\smile};\iota^{\smile});\iota \\
& = & \iota^{\smile};(\mathbb{I} \sqcup \iota;Q;Q^{\smile};\iota^{\smile});\iota \;=\; \iota^{\smile};\iota \sqcup \iota^{\smile};\iota;Q;Q^{\smile};\iota^{\smile};\iota \;=\; \iota^{\smile};\iota \sqcup Q;Q^{\smile} \\
& = & (\mathsf{ran}\,\Phi \to \mathsf{dom}\,Q) \sqcup Q;Q^{\smile}
\end{array}$$

In the same way, we obtain $\Psi;\Psi^{\smile} = \kappa^{\smile};\kappa \sqcup Q^{\smile};Q = (\mathsf{ran}\,\Xi \to \mathsf{ran}\,Q) \sqcup Q^{\smile};Q$. Finally, we obtain:

$$\mathrm{X}^{\smile};\mathrm{X} \sqcup \Psi^{\smile};\Psi = \mathrm{X}_0^{\smile};\iota;\iota^{\smile};\mathrm{X}_0 \sqcup \Psi_0^{\smile};\kappa;\kappa^{\smile};\Psi_0 = \mathrm{X}_0^{\smile};\mathrm{X}_0 \sqcup \Psi_0^{\smile};\Psi_0 = \mathbb{I} \qquad \square$$

## 5.7 Summary

In this chapter, we have presented component-free formalisations for the basic repertoire of the categorical approaches to graph transformation.

In particular, we achieved original component-free formulations for the existence condition for pullback complements, and for conflict-freeness and the original more intuitive single-pushout construction.

It is important that we have achieved all these formulations in a single, relational framework that makes no essential distinction between partial arrows and total arrows, or, by virtue of the self-duality of relation categories, between forward arrows and backward arrows.

As we have already seen in the different applications of the straight host construction, this enables combining the different approaches and re-using conditions and constructions in different contexts. An example for this is also the use of the straight and sloppy host constructions, that allowed to identify useful classes of restricting derivations — that approach had previously suffered from the apparent arbitrariness of the "tacked-on arrow" that inhibited automation and systematic application.

Approaches we have not covered are, to our knowledge, double-pullback transitions and the fibred approaches.

*Double-pullback transitions* were introduced by Heckel [Hec98] to supplement the double-pushout approach of systems modelling with a way to describe situations where simultaneously with the application of a specified rule, concurrent threads may change the context by, for example, applying unspecified rules to other parts of the graph, see also [HEWC97, EHTE97, EHL$^+$00].

The resulting double-pullback squares can be seen as a symmetric variant of restricting derivations, where an additional morphism is allowed after result construction, and certain restrictions are imposed on both sides. Translating the different kinds of restrictions from the double-pullback transition literature (like weak and horizontal injectivity wrt. commutative squares) into component-free form is a simple transfer exercise. It then becomes obvious that the "maximal pullback complement" of [EHL$^+$00, Prop. 3] is precisely our straight host construction (Def. 5.4.6), and the "lazy double-pullback transition" of [EHL$^+$00, Def. 6] is the construction of Theorem 5.4.11.

The *opfibration approach* of Banach [Ban93, Ban94] and the *fibred approach* of the present author [Kah96, Kah97] had as their main motivation applications in the domain of term graphs, where the "horizontal" arrows, representing rule application or deletion and addition, are intrinsically different from the "vertical" arrows representing rule matching.

Therefore, different categories needed to be identified: a substrate category $\mathcal{S}$ accommodating both horizontal and vertical morphisms, a subcategory $\mathcal{H}$ of $\mathcal{S}$, containing the horizontal morphisms, and a subcategory $\mathcal{V}$ of the arrow-category over $\mathcal{S}$, designating essentially the allowed vertical arrows, and offering the additional ability to impose constraints on whole commuting squares in $\mathcal{S}$. The approach then concentrates on the functor from $\mathcal{V}$ to $\mathcal{H}$ (for definitions concerning (op)cartesian arrows and (op)fibrations see for example [BW90, Chapt. 11]:

- opcartesian arrows replace those pushouts that produce results, for Banach in a modification of the single-pushout approach, and for our own in a modification of the double-pushout approach, and
- cartesian arrows play the rôle of the left-hand side pushout, including additional tacked-on arrows as in restricting derivations or "lazy double-pullback transitions".

Applications of this general framework crucially depend on the necessary instantiations of $\mathcal{S}$, $\mathcal{H}$, and $\mathcal{V}$.

The fibred approaches therefore do not offer themselves to a general relational treatment as easily as the standard categorical approaches. However, as we have seen for example in the context of straight and sloppy host constructions, the relational treatment allows considerable flexibility in adapting the categoric approaches, so it might be more interesting to *replace* applications of fibred approaches with fine-tuned relational approaches.

To a certain extent we are following this spirit when, in the next chapter, we build on the basis of the formalisations of this chapter to present an unprecedented amalgamation of the double-pushout and double-pullback approaches.

# Chapter 6

# Relational Rewriting in Dedekind Categories

In this chapter we show that it is possible to obtain a rule concept that combines the intuitive understandability of pushout rules with the replicative power of pullback rules. We achieve this by starting from the general setup of the double-pushout approach, and designating a part of the rule as *parameters*. It is actually sufficient to partition the gluing object into a *parameter part* and an *interface part*; the parameter and interface parts of the rule sides are then just the images of the parameter resp. interface parts of the gluing object; each rule side will in addition have a *context* outside the image of the rule morphisms.

The central idea is then to replace the double-pushout regime on the parameter part with a double-pullback regime, and to achieve this by just imposing appropriate constraints on *relational* morphisms.

In this way, all morphisms in the resulting double square diagram should essentially be total and univalent on interface and context parts, and injective and surjective on parameter parts. We showed a simple example of how this can work in the introduction, starting on page 22.

The first section of this chapter is devoted to the elaborating the issue of how the gluing object may be partitioned into parameter and interface parts, and how morphisms starting from the gluing object should respect this partitioning. In Sect. 6.2 we then define *pullouts* that can be understood as amalgamations of pushouts and pullbacks along the lines sketched above. The question of pullout complements is investigated in Sect. 6.3 — it turns out that for left-linear rules, the straight host construction continues to serve well.

Finally, we present one first possibility of adapting "double-pullout rewriting" towards slightly more general relational matchings in Sect. 6.5.

## 6.1 Gluing Setup

As mentioned above, we are going to present a generalisation of the pushout construction that incorporates elements of the behaviour of pullbacks. These pullback characteristics are needed for replication of the images of "variables" or *parameters*, while on the "non-variable" *interface* parts, the construction should behave essentially as a pushout.

This different treatment of two complementary parts is anchored at the rôle of the gluing object, and we therefore enrich the gluing object with a partial identity indicating the *interface*. The parameter part is then obtained as the semi-complement of the interface. This asymmetric setup has the advantage that the interface, which typically has a "border rôle", need not be coregular, while the parameter part typically has a "body rôle", which harmonises with the fact that, as a semi-complement, it is automatically coregular.

In graphs, this means that the interface may contain single nodes that are incident to variable edges, which is intuitively sensible, while it is less than clear what it should mean for a single node incident to an interface edge that it is at the same time a variable.

Throughout the following, we are working in a strict Dedekind category **D**.

**Definition 6.1.1** Let an object $G_0$ and a partial identity $u_0 : \mathsf{PId}\, G_0$ be given. The pair $(G_0, u_0)$ is then called a *gluing object* $G_0$ *along* $u_0$, and $u_0$ is called the *interface component* of $G_0$.

Furthermore, we define $v_0 := u_0^{\sim}$ and consider it as part of the gluing setup; we may now talk about a *gluing object* $G_0$ *along* $u_0$ *over* $v_0$, and we call $v_0$ the *parameter component* of $G_0$. □

The morphisms starting at the gluing object should *respect* this interface-parameter-setup in a way that is now to be made precise.

First of all, a morphism $\Phi : G_0 \to G_1$ starting at a gluing object along $u_0$ over $v_0$ gives rise to images $u_1$ and $v_1$ of the respective components. Then we demand that $\Phi$ "preserves" the interface rôle as far as possible; this means that the border between the parameter component of $G_1$ and its semi-complement should be contained in the interface component of $G_1$, that $\Phi$ is total on the interface part, and does not map any parameter items outside the border to interface parts. In addition, we have to demand that $\Phi$ is univalent on the pre-image of the parameter image border.

**Definition 6.1.2** [←135, 149, 185, 188, 190, 191] Let a gluing object $G_0$ along $u_0$ over $v_0$ and a morphism $\Phi$ from $G_0$ to another object $G_1$ be given.

In such a context, we shall use the following abbreviations: Let $u_1 := \mathsf{ran}\,(u_0{;}\Phi)$ and $v_1 := \mathsf{ran}\,(v_0{;}\Phi)$ be the images of the gluing components; let $b_0 := v_0 \sqcap u_0$ be the border between parameter part and interface part in the gluing object, and let $b_1 := v_1 \sqcap u_1$ be the border of the parameter image.

$\Phi$ is called *interface preserving* if the following conditions are satisfied:

(i) $v_1 \sqcap v_1^{\sim} \sqsubseteq u_1$

(ii) $\mathsf{dom}\,(\Phi{;}(v_1^{\sim} \sqcup u_1)) = u_0$

(iii) $b_1{;}\Phi^{\smile}{;}\Phi \sqsubseteq b_1$. □

This leads to many useful properties, most of which are "propagation of interface component" in other shapes:

**Lemma 6.1.3** [←144, 192]

(i) $(\mathsf{ran}\,\Phi)^{\sim} \sqcup u_1 = v_1^{\sim} \sqcup u_1$

(ii) $\mathsf{ran}\,\Phi \sqcap (\mathsf{ran}\,\Phi)^{\sim} \sqsubseteq u_1$

(iii) $u_1 \sqsubseteq v_1^{\sim} \sqcup b_1$

**Proof**:

(i) With Lemma 2.7.6.iii) we obtain:

$$(\mathsf{ran}\,\Phi)^{\sim} \sqcup u_1 = (u_1 \sqcup v_1)^{\sim} \sqcup u_1 = (v_1^{\sim} \smallsetminus u_1) \sqcup u_1 = v_1^{\sim} \sqcup u_1$$



(ii)
$$\begin{array}{rcll}
\mathsf{ran}\,\Phi \sqcap (\mathsf{ran}\,\Phi)^{\sim} & = & (u_1 \sqcup v_1) \sqcap (u_1 \sqcup v_1)^{\sim} & \\
& \sqsubseteq & (u_1 \sqcup v_1) \sqcap v_1^{\sim} & \text{Lemma 2.7.3.ii)} \\
& \sqsubseteq & u_1 \sqcup (v_1 \sqcap v_1^{\sim}) & \\
& \sqsubseteq & u_1 \sqcup u_1 & \text{(i)} \\
& = & u_1 &
\end{array}$$

(iii) $v_1^{\sim} \sqcup b_1 = v_1^{\sim} \sqcup (v_1 \sqcap u_1) = (v_1^{\sim} \sqcup v_1) \sqcap (v_1^{\sim} \sqcup u_1) = \mathbb{I} \sqcap (v_1^{\sim} \sqcup u_1) = v_1^{\sim} \sqcup u_1$ □

The starting point for our generalisation of pushouts is then a span of interface preserving relations starting from a common gluing object.

**Definition 6.1.4** A *gluing setup* $(G_0, u_0, v_0, \Xi, \Phi)$ consists of a gluing object $G_0$ along $u_0$ over $v_0$, and a span of interface-preserving morphisms $G_2 \xleftarrow{\Xi} G_0 \xrightarrow{\Phi} G_1$, where $G_2$ is called the *host object* and $G_1$ is called the *rule side*.

In a gluing setup with names as above, we additionally define the following names for the interface, parameter, and target specific components in the target objects:

$$\begin{array}{rclrclrcl}
v_1 & := & \mathsf{ran}\,(v_0{;}\Phi) & u_1 & := & \mathsf{ran}\,(u_0{;}\Phi) & r_1 & := & (\mathsf{ran}\,\Phi)^{\sim} \\
v_2 & := & \mathsf{ran}\,(v_0{;}\Xi) & u_2 & := & \mathsf{ran}\,(u_0{;}\Xi) & h_2 & := & (\mathsf{ran}\,\Xi)^{\sim}
\end{array}$$

In addition, we define names for the borders between the interface and parameter parts, and for the whole non-parameter parts:

$$\begin{array}{rclrclrcl}
b_1 & := & u_1 \sqcap v_1 & c_1 & := & u_1 \sqcup r_1 & & & \\
b_2 & := & u_2 \sqcap v_2 & c_2 & := & u_2 \sqcup h_2 & q_2 & := & b_2 \sqcup h_2
\end{array}$$ □

The next step will now be to complete such a gluing setup to a commuting square of the following shape:

$$\begin{tikzcd}
G_0 \arrow[r, "\Phi"] \arrow[d, "\Xi"'] & G_1 \arrow[d, "X"] \\
G_2 \arrow[r, "\Psi"] & G_3
\end{tikzcd}$$

In order to enable relational rewriting with parameters, this square is to include characteristics of the pushout of forward components of the arrows restricted to the interface and context parts, and of the pullback of backward components of the arrows restricted to the parameter parts.

In the next section we directly perform such an amalgamation of the relational variants of pushout and pullback. Since this is still very close to the categorical setup of mappings, it has certain shortcomings from a relational rewriting point of view, so we present a "more relational" variant in Sect. 6.5.

## 6.2 Amalgamating Pushouts and Pullbacks to Pullouts

A simple first approach to completing a gluing setup to a full square starts with producing a pushout for the interface parts $u_0;\Phi$ and $u_0;\Xi$, and a pullback for the converses of the parameter parts, i.e., for $\Phi^{\smile};v_0$ and $\Xi^{\smile};u_0$. To make these pushouts and pullbacks possible, $\Phi$ and $\Xi$ have to behave "essentially like mappings" in the forward direction on the interface part, and in the backward direction on the parameter part:

**Definition 6.2.1** [←180, 182] Given a gluing object $G_0$ along $u_0$, a morphism $\Phi : G_0 \leftrightarrow G_1$ is called a *standard gluing morphism* iff the following conditions are satisfied:

(i) $\Phi$ is interface preserving,

(ii) $\Phi$ is total on $u_0$, that is, $u_0 \sqsubseteq \mathsf{dom}\,\Phi$,

(iii) $\Phi$ is univalent on $u_0$, that is, $u_0;\Phi$ is univalent, and

(iv) $\Phi$ is almost-injective besides $u_0$. □

Once we have the pushout of the interface part and the pullback of the parameter part, the two result objects then have to be glued together along a gluing relation induced by the restrictions of the pushout and pullback morphisms to the borders $b_1$ and $b_2$.

The following diagram should help with orientation:

$$\begin{tikzcd}
& u_0 \arrow[rrrr, leftrightarrow, "u_0 \sqcap v_0"] \arrow[dl, "u_0;\Xi"'] \arrow[dr, "u_0;\Phi"] & & & & v_0 & \\
c_2 \arrow[dr, "c_2;\Psi"'] & \text{PO} & c_1 \arrow[dl, "c_1;\mathrm{X}"] & & v_2 \arrow[ur, "\Xi^{\smile};v_0"] & \text{PB} & v_1 \arrow[ul, "\Phi^{\smile};v_0"'] \\
& c_3 \arrow[ddrr, "\Lambda_u"'] & & & & v_3 \arrow[llll, "B"'] \arrow[ul, "\Psi^{\smile};v_2"] \arrow[ur, "\mathrm{X}^{\smile};v_1"'] \arrow[ddll, "\Lambda_v"] & \\
& & & \text{gluing} & & & \\
& & & G_3 & & &
\end{tikzcd}$$

According to this intuition, the specification of this construction also has to join the specifications of pushout and pullback into a single specification. We have to be careful to restrict the reflexive transitive closures to those parts of $G_1$ and $G_2$ that are governed by the pushout construction, since otherwise the reflexive part would override the pullback domains. The final gluing is not directly reflected in these conditions since interface preservation keeps the border part inside the interface part, so the gluing components for the interface part already cover the final gluing.



**Definition 6.2.2** Given a gluing setup $G_1 \xleftarrow{\Phi} G_0 \xrightarrow{\Xi} G_2$ along $u_0$ over $v_0$ in a Dedekind category $\mathbf{D}$, a cospan $G_1 \xrightarrow{\mathrm{X}} G_3 \xleftarrow{\Psi}$ of relations in $\mathbf{D}$ is a *pullout for* $G_1 \xleftarrow{\Phi} G_0 \xrightarrow{\Xi} G_2$ *along* $u_0$ iff the following conditions are satisfied:

$$\begin{array}{rcl@{\quad}c@{\quad}c}
\Phi;\mathrm{X} & = & \Xi;\Psi & & \\
\mathrm{X};\Psi^{\smile} & = & (\Phi^{\smile};u_0;\Xi)^{\boxast} & \sqcup & \Phi^{\smile};v_0;\Xi \\
\mathrm{X};\mathrm{X}^{\smile} & = & c_1;(\Phi^{\smile};u_0;\Xi)^{\triangleright};c_1 & \sqcup & \mathsf{dom}\,(\Phi^{\smile};v_0;\Xi) \\
\Psi;\Psi^{\smile} & = & c_2;(\Phi^{\smile};u_0;\Xi)^{\triangleleft};c_2 & \sqcup & \mathsf{ran}\,(\Phi^{\smile};v_0;\Xi) \\
\mathbb{I} & = & \mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi & \sqcup & (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi)
\end{array}$$ □

From the shape of these conditions it is obvious that most of the details about $\Phi$ and $\Xi$ are completely irrelevant, and we only need access to the resulting parameter and interface parts. In the spirit of the definitions of tabulations and gluings, we therefore can abstract away from $\Phi$ and $\Xi$, and substitute

$$U := \Phi^{\smile};u_0;\Xi \ , \qquad V := \Phi^{\smile};v_0;\Xi \ .$$

With this, the essential effects of interface preservation are the following consequences of Def. 6.1.2.ii):

$$\begin{array}{rcl}
c_1;V & = & c_1;\Phi^{\smile};v_0;\Xi \ = \ c_1;\Phi^{\smile};u_0;v_0;\Xi \ \sqsubseteq \ U \\
V;c_2 & = & \Phi^{\smile};v_0;\Xi;c_2 \ = \ \Phi^{\smile};v_0;u_0;\Xi;c_2 \ \sqsubseteq \ U
\end{array}$$

Reflecting the fact that in the pullback component, $\Phi^{\smile};v_0$ and $\Xi^{\smile};v_0$ need not be surjective on $v_0$, we have to take the possibility into account that domain and range of $V$ are strictly included in $v_1$ respectively $v_2$. Therefore, we still have to supply the partitioning into parameter part and non-parameter part as additional parameter. As for the gluing setup, we take the non-parameter part as primitive; here, this comprises not only the interface part, but also the context parts, that is, the parts corresponding to $(\mathsf{ran}\,\Phi)^{\sim}$ and $(\mathsf{ran}\,\Xi)^{\sim}$.

**Definition 6.2.3** [←136, 137] Let two relations $U, V : G_1 \leftrightarrow G_2$ and two partial identities $c_1 : \mathsf{PId}\,G_1$ and $c_2 : \mathsf{PId}\,G_2$ be given. Define:

$$v_1 := c_1^{\sim} \ , \qquad v_2 := c_2^{\sim} \ .$$

If we have *direct interface preservation*:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\mathsf{dom}\,U & \sqsubseteq & c_1 & \mathsf{dom}\,V & \sqsubseteq & v_1 & c_1;V & \sqsubseteq & U \\
\mathsf{ran}\,U & \sqsubseteq & c_2 & \mathsf{ran}\,V & \sqsubseteq & v_2 & V;c_2 & \sqsubseteq & U \ ,
\end{array}$$

then a cospan $G_1 \xrightarrow{\mathrm{X}} G_3 \xleftarrow{\Psi} G_2$ is called a *glued tabulation for* $V$ *along* $U$ *on* $c_1$ *and* $c_2$ iff the following conditions hold:

$$\begin{array}{rcl@{\quad}c@{\quad}c}
\mathrm{X};\Psi^{\smile} & = & U^{\boxast} & \sqcup & V \\
\mathrm{X};\mathrm{X}^{\smile} & = & c_1;U^{\triangleright};c_1 & \sqcup & \mathsf{dom}\,V \\
\Psi;\Psi^{\smile} & = & c_2;U^{\triangleleft};c_2 & \sqcup & \mathsf{ran}\,V \\
\mathbb{I} & = & \mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi & \sqcup & (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi)
\end{array}$$ □

The main advantage of this more abstract formulation is that it is more concise, and that fewer components and conditions have to be dealt with in proofs.

Since the setup of the glued tabulation is implied by the gluing setup, the following theorem also shows monomorphy of pullouts.

**Theorem 6.2.4** [←184] The characterisation of glued tabulations is monomorphic.

**Proof**: In the context of Def. 6.2.3, let $G_1 \xrightarrow{\mathrm{X}'} G_4 \xleftarrow{\Psi'}$ be a second glued tabulation for $V$ along $U$ on $c_1$ and $c_2$. Define:

$$\mathrm{Y} := \mathrm{X}^{\smile};c_1;\mathrm{X}' \sqcup \Psi^{\smile};c_2;\Psi' \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X}' \sqcap \Psi^{\smile};v_2;\Psi')$$

The key tools are the first three glued tabulation conditions, which imply:

$$\mathrm{X};\Psi^{\smile} = \mathrm{X}';\Psi'^{\smile} \qquad \mathrm{X};\mathrm{X}^{\smile} = \mathrm{X}';\mathrm{X}'^{\smile} \qquad \Psi;\Psi^{\smile} = \Psi';\Psi'^{\smile}$$

Factorisation: Let $v_4 := \mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi$ and $c_4 := \mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi$, then:

$$\begin{array}{lll}
& \mathrm{X};\mathrm{Y} & \\
= & \mathrm{X};\mathrm{X}^{\smile};c_1;\mathrm{X}' \sqcup \mathrm{X};\Psi^{\smile};c_2;\Psi' \sqcup \mathrm{X};(\mathrm{X}^{\smile};v_1;\mathrm{X}' \sqcap \Psi^{\smile};v_2;\Psi') & \\
= & \mathrm{X};\mathrm{X}^{\smile};c_1;\mathrm{X}' \sqcup \mathrm{X};\Psi^{\smile};c_2;\Psi' \sqcup v_1;\mathrm{X};(\mathrm{X}^{\smile};v_1;\mathrm{X}' \sqcap \Psi^{\smile};v_2;\Psi') & \mathrm{X}\text{ almost-inj. bes. } c_1 \\
= & \mathrm{X}';\mathrm{X}'^{\smile};c_1;\mathrm{X}' \sqcup \mathrm{X}';\Psi'^{\smile};c_2;\Psi' \sqcup (v_1;\mathrm{X}' \sqcap v_1;\mathrm{X};\Psi^{\smile};v_2;\Psi') & \mathrm{X}\text{ almost-inj. bes. } c_1 \\
= & \mathrm{X}';c_4 \sqcup (v_1;\mathrm{X}' \sqcap v_1;\mathrm{X}';\Psi'^{\smile};v_2;\Psi') & \\
= & \mathrm{X}';c_4 \sqcup v_1;\mathrm{X}';(\mathrm{X}'^{\smile};v_1;\mathrm{X}' \sqcap \Psi'^{\smile};v_2;\Psi') & \mathrm{X}'\text{ almost-inj. bes. } c_1 \\
= & \mathrm{X}';c_4 \sqcup v_1;\mathrm{X}';v_4 & \\
= & \mathrm{X}' &
\end{array}$$

In the same way, we also obtain $\Psi;\mathrm{Y} = \Psi'$. This helps with univalence:

$$\begin{array}{rcl}
\mathrm{Y}^{\smile};\mathrm{Y} & = & \mathrm{X}'^{\smile};c_1;\mathrm{X};\mathrm{Y} \sqcup \Psi'^{\smile};c_2;\Psi;\mathrm{Y} \sqcup (\mathrm{X}'^{\smile};v_1;\mathrm{X} \sqcap \Psi'^{\smile};v_2;\Psi);\mathrm{Y} \\
& \sqsubseteq & \mathrm{X}'^{\smile};c_1;\mathrm{X}' \sqcup \Psi'^{\smile};c_2;\Psi' \sqcup (\mathrm{X}'^{\smile};v_1;\mathrm{X};\mathrm{Y} \sqcap \Psi'^{\smile};v_2;\Psi;\mathrm{Y}) \\
& = & \mathrm{X}'^{\smile};c_1;\mathrm{X}' \sqcup \Psi'^{\smile};c_2;\Psi' \sqcup (\mathrm{X}'^{\smile};v_1;\mathrm{X}' \sqcap \Psi'^{\smile};v_2;\Psi') \\
& = & \mathbb{I}
\end{array}$$

The following fills in the gap towards surjectivity:

$$\begin{array}{rcl}
\mathsf{ran}\,(\mathrm{X}^{\smile};v_1;\mathrm{X}' \sqcap \Psi^{\smile};v_2;\Psi') & = & \mathsf{ran}\,(v_1;\mathrm{X}' \sqcap \mathrm{X};\Psi^{\smile};v_2;\Psi') \\
& = & \mathsf{ran}\,(v_1;\mathrm{X}' \sqcap \mathrm{X}';\Psi'^{\smile};v_2;\Psi') \\
& = & \mathsf{ran}\,(\mathrm{X}'^{\smile};v_1;\mathrm{X}' \sqcap \Psi'^{\smile};v_2;\Psi') \\
& = & v_4
\end{array}$$

Since for $Y^{\smile}$ the same argument is valid, too, Y is a bijective mapping. □



Adapting the above sketch of a construction to this abstract setting is straightforward. The only minor technical complication arises from the restriction of the non-parameter gluing to $c_1$ and $c_2$ which is necessary for preserving the possibility that the parameter parts of X and $\Psi$ may be partial. For achieving this cleanly, we have to construct this first gluing from appropriate subobjects. For the parameter part, however, no such subobjects are necessary since the tabulation does not reach beyond the domain and range of $V$.

**Definition 6.2.5** [←141, 142, 173] Under the preconditions of Def. 6.2.3, a cospan $G_1 \xrightarrow{\mathrm{X}} G_3 \xleftarrow{\Psi} G_2$ is a *constructed pullout* if it may be obtained by the following steps:

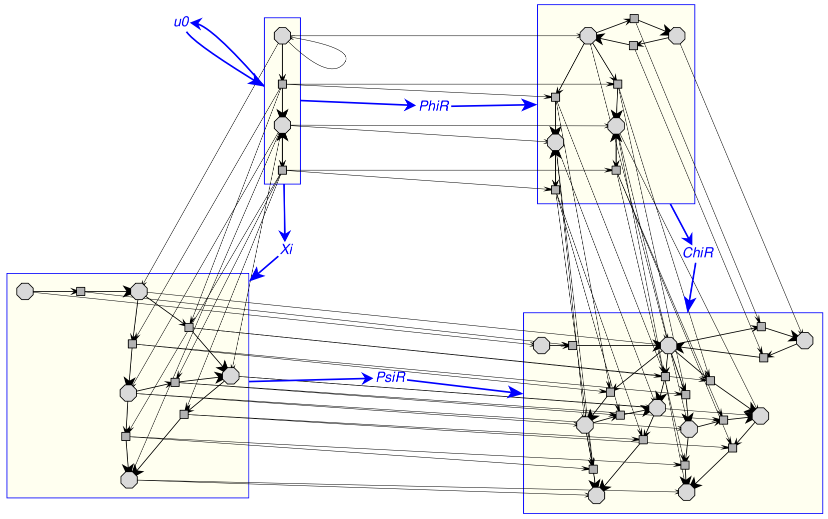
- Let $\lambda_1 : C_1 \rightarrowtail G_1$ be a subobject for $c_1$, and let $\lambda_2 : C_2 \rightarrowtail G_2$ be a subobject for $c_2$.
- Let $C_1 \xrightarrow{\mathrm{X}_u} G_u \xleftarrow{\Psi_u} C_2$ be a gluing for $\lambda_1;U;\lambda_2^{\smile}$.
- Let $G_1 \xleftarrow{\mathrm{X}_v} G_v \xrightarrow{\Psi_v} G_2$ be a tabulation for $V$.
- Define $B := \mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqcup \Psi_v;\lambda_2^{\smile};\Psi_u$.
- Let $G_v \xrightarrow{\Lambda_v} G_3 \xleftarrow{\Lambda_u} G_u$ be a gluing for $B$.
- Define: $\mathrm{X} := \lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \mathrm{X}_v^{\smile};\Lambda_v$ $\qquad \Psi := \lambda_2^{\smile};\Psi_u;\Lambda_u \sqcup \Psi_v^{\smile};\Lambda_v$ □

$$\begin{tikzcd}
C_2 \arrow[dr, "\Psi_u"'] & & C_1 \arrow[ll, "\lambda_1;U;\lambda_2^{\smile}"'] \arrow[dl, "\mathrm{X}_u"] \arrow[ll, phantom, shift right=4ex, "\text{gluing}"] & & v_2 & & v_1 \arrow[ll, "V"'] \arrow[ll, phantom, shift right=4ex, "\text{tabulation}"] \\
& G_u \arrow[drr, "\Lambda_u"'] & & & & G_v \arrow[llll, "B"'] \arrow[ul, "\Psi_v"'] \arrow[ur, "\mathrm{X}_v"'] \arrow[dll, "\Lambda_v"] \arrow[llll, phantom, shift left=5ex, "\text{gluing}"] & \\
& & & G_3 & & &
\end{tikzcd}$$

The stacked gluing construction induces large expressions, so proving the correctness of this construction involves lengthy tedious calculations which are relegated to Appendix B.1 starting on page 173; these calculations show the following:

**Theorem 6.2.6** [←173] A constructed pullout is a well-defined glued tabulation. □

We have already shown an example pullout in the introduction (page 24); we now show a slight variation that has non-empty contexts in $\mathcal{R}$ and $\mathcal{H}$.




We still have to show that constructed pullout for a gluing setup with standard gluing morphisms is well-defined, and produces a commuting square:

**Theorem 6.2.7** [←177] Let a gluing setup $(G_0, u_0, v_0, \Xi, \Phi)$ be given where $\Phi$ and $\Xi$ both are standard gluing morphisms. Defining $U := \Phi^{\smile};u_0;\Xi$ and $V := \Phi^{\smile};v_0;\Xi$ then ensures direct interface preservation, and a constructed pullout $G_1 \xrightarrow{X} G_3 \xleftarrow{\Psi} G_2$ commutes, that is, $\Phi;X = \Xi;\Psi$.

**Proof**: Direct interface preservation follows from interface preservation as we have seen on page 135. The proof of commutativity may be found in Appendix B.1 on page 177. □

## 6.3 Pullout Complements

For pullout complements, let us first consider the case that $v_0;\Phi$ is univalent, that is, that the rule is left-linear. For the pullback-part of the construction, Theorem 5.5.5 is then applicable and the parameter image may just be copied from the application graph to the host graph and does not need any special treatment. As a result, we may use the straight host construction of Def. 5.4.6 to obtain a pullout complement, as proven in Sect. B.2:

**Theorem 6.3.1** [←139, 140, 142, 150, 179, 183, 200] Let a gluing object $G_0$ along $u_0$ over $v_0$ and a standard gluing morphism $\Phi : \mathcal{G} \to \mathcal{L}$ be given, and a morphism $X : \mathcal{L} \to \mathcal{A}$. Define:

$$\begin{array}{lll} v_1 := \mathsf{ran}\,(v_0;\Phi) & u_1 := \mathsf{ran}\,(u_0;\Phi) & c_1 := u_1 \sqcup (\mathsf{ran}\,\Phi)^{\sim} \\ v_3 := \mathsf{ran}\,(v_1;X) & u_3 := \mathsf{ran}\,(u_1;X) & c_3 := u_3 \sqcup (\mathsf{ran}\,X)^{\sim} \end{array}$$

If the following conditions are satisfied:

(i) $\Phi$ is univalent

(ii) $c_1;X$ is univalent, and $c_1 \sqsubseteq \mathsf{dom}\,X$,



(iii) X is almost-injective besides $u_1$,

(iv) $X;(\mathsf{ran}\,X)^{\sim} \sqsubseteq u_1;X$,

then the straight host construction of Def. 5.4.6 delivers a pullout complement.

Furthermore, $\Xi$ is a standard gluing morphism if the following additional conditions are satisfied:

(v) $v_3 \sqcap v_3^{\sim} \sqsubseteq u_3$ and

(vi) $X;u_3 \sqsubseteq u_1;X$. □

In the general case, however, we need the same equivalence relation complement as for general pullback complements, only in the appropriate restriction to the parameter part. In the preconditions of the following theorem, the only differences to the preconditions of Theorem 6.3.1 are the omission of 6.3.1.i), so that now, by $\Phi$ being a standard gluing morphism, only $u_0;\Phi$ needs to be univalent instead of the whole of $\Phi$, and insertion of a new condition (iv).

**Theorem 6.3.2** [←140, 142, 143, 181, 182] Let a gluing object $\mathcal{G}$ along $u_0$ over $v_0$ and a standard gluing morphism $\Phi : \mathcal{G} \to \mathcal{L}$ be given, and a morphism $X : \mathcal{L} \to \mathcal{A}$. Define:

$$\begin{array}{lll} v_1 := \mathsf{ran}\,(v_0;\Phi) & u_1 := \mathsf{ran}\,(u_0;\Phi) & c_1 := u_1 \sqcup (\mathsf{ran}\,\Phi)^{\sim} \\ v_3 := \mathsf{ran}\,(v_1;X) & u_3 := \mathsf{ran}\,(u_1;X) & c_3 := u_3 \sqcup (\mathsf{ran}\,X)^{\sim} \end{array}$$

If the following conditions are satisfied:

(i) $c_1;X$ is univalent, and $c_1 \sqsubseteq \mathsf{dom}\,X$,

(ii) X is almost-injective besides $u_1$,

(iii) $X;(\mathsf{ran}\,X)^{\sim} \sqsubseteq u_1;X$,

(iv) there is a partial equivalence relation $\Theta : \mathcal{A} \leftrightarrow \mathcal{A}$ such that

$$\begin{array}{lll} \mathsf{ran}\,\Theta = v_3 & & X^{\smile};v_1;X \sqcap \Theta \sqsubseteq \mathbb{I} \\ u_3;\Theta \sqsubseteq u_3 & & v_1;X;\Theta = \Phi^{\smile};v_0;\Phi;X \ , \end{array}$$

then there is a pullout complement $\mathcal{A} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi} \mathcal{L}$ constructed as follows:

- Let $\mathcal{V}$ contain the variable instantiation, characterised as a combined quotient and subobject by $\nu : \mathcal{A} \leftrightarrow \mathcal{V}$ with:

$$\nu^{\smile};\nu = \mathbb{I} \ , \qquad \nu;\nu^{\smile} = \Theta \ .$$

- Let $\mathcal{C}$ be the subobject of $\mathcal{A}$ containing the context:

$$\lambda : \mathcal{C} \leftrightarrow \mathcal{A} \qquad \lambda;\lambda^{\smile} = \mathbb{I} \qquad \lambda^{\smile};\lambda = c_3$$

- Let $\mathcal{C} \xrightarrow{\iota} \mathcal{H} \xleftarrow{\kappa} \mathcal{V}$ be the gluing of $Z : \mathcal{C} \leftrightarrow \mathcal{V}$ with $Z := \lambda;\nu$
- Define: $\Psi := \iota^{\smile};\lambda \sqcup \kappa^{\smile};\nu^{\smile}$ and $\Xi := \Phi;X;\Psi^{\smile}$.

$$\begin{tikzcd}
\mathcal{L} \arrow[d, "\mathrm{X}"'] & & \mathcal{G} \arrow[ll, "\Phi"'] \arrow[d, "\Xi"] \\
\mathcal{A} \arrow[dr, "\nu"] & & \mathcal{H} \arrow[ll, "\Psi"'] \\
\mathcal{S} \arrow[u, "\lambda"] \arrow[r, "Z"'] \arrow[urr, "\iota"] & \mathcal{V} \arrow[ur, "\kappa"'] &
\end{tikzcd}$$

Furthermore, $\Xi$ is a standard gluing morphism if the following additional conditions are satisfied:

(v) $v_3 \sqcap v_3^{\smile} \sqsubseteq u_3$ and

(vi) $\mathrm{X};u_3 \sqsubseteq u_1;\mathrm{X}$. □

The proof of this theorem may be found in Sect. B.2, starting on page 181.

## 6.4 Pullout Rewriting

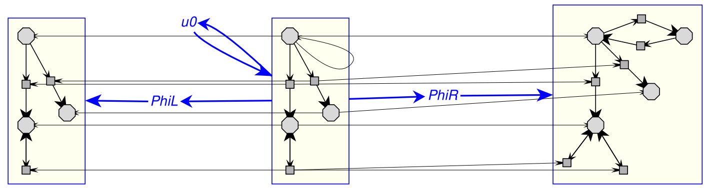
With pullouts and pullout complements in place, it is straightforward to set up a *double-pullout approach* in analogy to the double-pushout approach.

**Definition 6.4.1** In a Dedekind category **D**, a *double-pullout rule* $\mathcal{L} \xleftarrow{\Phi_L} (\mathcal{G}, u_0) \xrightarrow{\Phi_R} \mathcal{R}$ consists of a gluing object $\mathcal{G}$ along $u_0$, the left-hand side and right-hand side objects $\mathcal{L}$ and $\mathcal{R}$, and two standard gluing morphisms $\Phi_L : \mathcal{G} \leftrightarrow \mathcal{L}$ and $\Phi_R : \mathcal{G} \leftrightarrow \mathcal{R}$.

Such a double-pullout rule is called *left-linear* iff $\Phi_L$ is univalent.

Given such a rule, if for an application object $\mathcal{A}$ and a morphism $\mathrm{X}_L : \mathcal{L} \leftrightarrow \mathcal{A}$ all preconditions of Theorem 6.3.2 hold, then the rule is *applicable to* $\mathcal{A}$ *via* $\mathrm{X}_L$, and *application* first calculates a pullout complement $\mathcal{G} \xrightarrow{\Xi} \mathcal{H} \xrightarrow{\Psi_L} \mathcal{A}$ for $(\mathcal{G}, u_0) \xrightarrow{\Phi_L} \mathcal{L} \xrightarrow{\mathrm{X}_L} \mathcal{A}$, and then constructs a pullout $\mathcal{R} \xrightarrow{\mathrm{X}_R} \mathcal{B} \xleftarrow{\Psi_R} \mathcal{H}$ for $\mathcal{R} \xleftarrow{\Phi_R} (\mathcal{G}, u_0) \xrightarrow{\Xi} \mathcal{H}$, where $\mathcal{B}$ is the *result* of the application. □

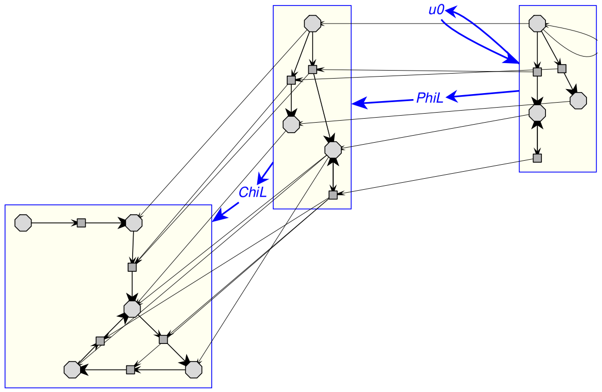
The following left-linear rule will be used below:




The condition that $\mathrm{X}_L$ is almost-injective besides $\mathsf{ran}\,(u_0;\Phi_L)$, used as precondition 6.3.1.iii) and 6.3.2.ii) in the pullout complement constructions, is in fact an *extended identification condition* that forbids identifications via $\mathrm{X}_L$ not only in the context $(\mathsf{ran}\,\Phi_L)^{\smile}$, but also in

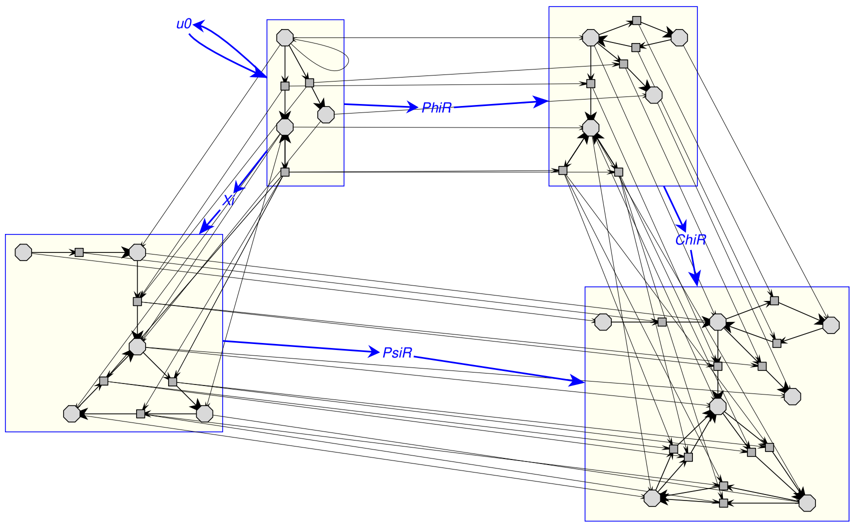


the parameter part $\mathsf{ran}\,(v_0;\Phi_L)$ of the rule's left-hand side. If this extended identification condition is violated, the parameter part of the resulting host morphism is not necessarily (almost-)injective anymore. This happens for example in the following setup:



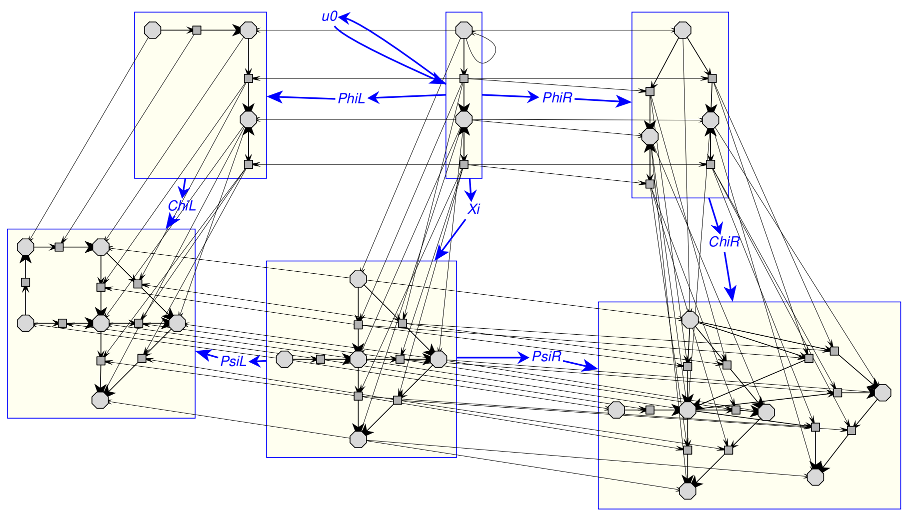

Here, $\Phi_L$ is an identity, and the straight host construction also lets $\Psi_L$ be an identity and $\Xi = \mathrm{X}_L$, where the parameter part is not injective.

For the right-hand side, the pullout *construction* of Def. 6.2.5 continues to be possible, but it does not produce a pullout:




Commutativity fails: $\Xi;\Psi_R$ relates both "strands" of the parameter part of the gluing graph with their joint image in the result graph, while $\Phi_R;\mathrm{X}_R$ relates each "strand" only with its own image.

In the Theorems 6.3.1 and 6.3.2, additional conditions ensure that the resulting host morphism $\Xi$ will be a standard gluing morphism. These conditions, together with the original dangling condition that here occurs as 6.3.1.iv) and 6.3.2.iii), give rise to an *extended dangling condition* that forbids edges between the context of the application graph and not only the image of that part of the left-hand side not covered by $\Phi_L$, but also the image of the parameter part of the left-hand side. In the following example, this last condition fails: the preserved context edge is incident with a parameter node outside the border:




The pullout construction of Def. 6.2.5 is still possible, but again does not deliver a pullout: this time it is the equation for $X_R;X_R^{\smile}$ that fails, since the context edge "inhibits replication" of its incident node, so $X_R$ is not almost-injective besides $u_1$.

Since with $u_0 = \mathbb{L}$, the double-pullout approach reduces to the double-pullback approach, it inherits all the expressivity of the double-pullback approach, which comprises that of most well-known graph rewriting systems, see [BJ01, JKH00]. On the other hand, with $u_0 = \mathbb{I}$ the double-pullout approach reduces to the double-pushout approach, so it can be employed in contexts where this more popular approach was employed, with "full backwards compatibility".

Already the simple rules we have shown so far should give some understanding how the double-pullout approach allows to harness the replicative power of the pullback approaches inside the intuitive setting of the double-pushout approach. The way parameters are handled is, due to its relation with pullback rewriting, of course quite different from more substitution-based variable concepts, but still reasonably easy to grasp. When emulating hyperedge substitution, it is probably most natural to do this in the context of hypergraphs considered as sigDHG-algebras.



Apart from this "standard" double-pullout rewriting approach, other approaches are of course possible. Certain constrained variants of the restricted derivations approach may seem most attractive — the following all tackle restrictions imposed by the (extended) dangling condition:

- Use the straight host construction (or the sloppy host construction) in all cases where it produces a host morphism that is a standard gluing morphism.
- Adapt the straight host construction so that it deletes edges that are incident with the parameter part ($\mathsf{ran}\,\Psi = (\mathsf{ran}\,X \to \mathsf{ran}\,(u_0;\Phi;X)) \sqcup \mathsf{ran}\,(v_0;\Phi;X)$), and then proceed as above.
- Analogously adapt the sloppy host construction, or the construction of Theorem 6.3.2, which is a generalisation of it, so that it deletes edges that are incident with the parameter part.

The fact that it is possible to exert such fine-grained control over the rewriting mechanism while staying on the component-free level all the time is clearly a major advantage of the relation-algebraic approach.

## 6.5 The Weak Pullout Construction

The pullout construction quite obviously cannot hide its genesis from constructions intended for mappings.

In particular, the result morphisms X and $\Psi$ are univalent on the interface part and (besides border identification) injective on the parameter part. The latter implies that overlaps between the images of different parameters forbid application of double-pullout rules, and applying the constructions nevertheless will not preserve this overlap, as has been seen in the example on page 141.

Since such a limitation may not always be acceptable, we now explore a variant that allows slightly more general relations. In particular, we are going to allow identifications between the images of different parameters as prescribed by the host, and right-hand side morphisms that are not univalent on the interface part, both, however, with certain restrictions, that we are now going to introduce.

First of all, let us consider what kinds of gluing setups we would like to start with.

According to the discussion above, we will not require that the parameter part of $X_L$ be injective. In comparison with the "pure" double-pullout approach, this means that we also drop univalence of the converse of the parameter part of the host morphism $\Xi$. However, we preserve totality and univalence of the interface part, since for host morphisms, the interface should essentially be embedded as-is, and we formulate this as univalence.

**Definition 6.5.1** Let a gluing object $G_0$ along $u_0$ over $v_0$ and an interface preserving morphism $\Xi$ from $G_0$ to another object $G_2$ be given.

$\Xi$ is called a *host morphism* along $u_0$ over $v_0$ if the following condition holds:

- *interface univalence*: $u_0;\Xi$ is univalent □

As another motivation one can argue that for parameters, the host morphism specifies *instantiation*, and therefore need not be restricted.

The morphism from the gluing object to a rule's right-hand side specifies *replication* of parameters, and therefore must not identify different parameters, since this would involve unification of their instantiation, which will in general not be possible. Therefore, we are going to forbid such identification. However, we cannot enforce this via injectivity on the parameter component, since we still want to allow the right-hand side morphism to specify identification of variable borders. Therefore, almost-injectivity is the natural choice. The difference with the functional setting is that we do not prescribe univalence on the interface part.

**Definition 6.5.2** Let a gluing object $G_0$ along $u_0$ over $v_0$ and an interface preserving morphism $\Phi$ from $G_0$ to another object $G_1$ be given.

$\Phi$ is called an *rhs-morphism* along $u_0$ over $v_0$ if the following condition holds:

- *parameter injectivity*: $\Phi$ is almost-injective (on $v_0$) besides $u_0$. □

Putting these two together, we obtain a special kind of span that is going to be the starting point for our result construction:

**Definition 6.5.3** A gluing setup $(G_0, u_0, v_0, \Xi, \Phi)$ is called a *result gluing setup* if $\Xi$ is a host morphism along $u_0$ over $v_0$ and $\Phi$ is an rhs-morphism along $u_0$ over $v_0$. □

The following figure now summarises a few important aspects about our desired result construction.

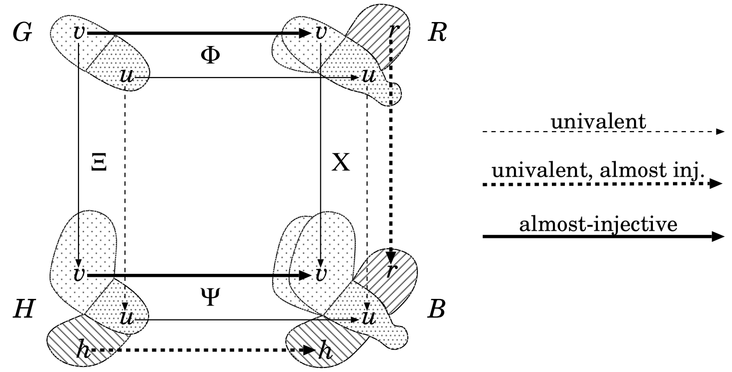
We start from a gluing object over a gluing graph $G$, with interface component $u$ and parameter component $v$, a host morphism $\Xi : G \to H$ from $G$ to a host graph $H$, and an rhs-morphism $\Phi : G \to R$ from $G$ to a right-hand side $R$.

Because of interface preservation (see Lemma 6.1.3.ii), the parts $h_2 := (\mathsf{ran}\,\Xi)^\sim$ and $r_1 := (\mathsf{ran}\,\Phi)^\sim$ can be attached to the respective image of $G$ only at the interface parts — including of course the border between interface and parameter parts, but this is not made very explicit in the drawing.

The parameter part of the host graph $H$ will contain what is considered as an *instantiation* of the parameter component of $G$, while the parameter part of $R$ may indicate the need for *replication* of the parameters — this should explain the different graphical effects used for the parameter parts of $H$ and $R$.

The interface part of $H$ has to be a univalent image of the interface component of $G$, but it may contain identifications — the possibility of identifications is nowhere indicated in the drawing. In contrast, the interface part of $R$ will be considered as an instantiation of the interface component of $G$, and is not subject to any particular constraints, except those imposed by interface preservation, mostly concerning the border with the other parts. (In practical applications, the interface component will frequently be discrete and therefore only designate legitimate borders; the scope of possible instantiations via $\Phi$ is then of course restricted to identifications.)






The result graph $B$ then should contain copies of $r$ and $h$, which may, however, suffer identification of different parts of their borders, therefore we can demand only almost-injectivity for the result morphisms on these parts.

Almost the same applies for the interface part of $R$, but there we cannot even demand almost-injectivity, since the host morphism $\Xi$ may dictate internal identifications in the interface part, which then carry over to the result.

For the parameter part of the host graph $H$, we do not just need a single copy, but a "replicated copy", according to the replication prescription in the parameter part of $\Phi$.

We need to impose an additional condition that slightly restricts the liberties of the result gluing setups of Def. 6.5.3 in order to ensure well-definedness of the result construction:

**Definition 6.5.4** [←195] A gluing setup $(G_0, u_0, v_0, \Xi, \Phi)$ is called *reasonable* if the whole of $G_0$ is covered by the domains of the univalent part of $\Phi$ and of the injective part of $\Xi$:

$$\mathbb{I} \sqsubseteq \mathsf{dom}\,(\mathsf{upa}\,\Phi) \sqcup \mathsf{injdom}\,\Xi\ , \qquad (\mathsf{spurious})$$

and if

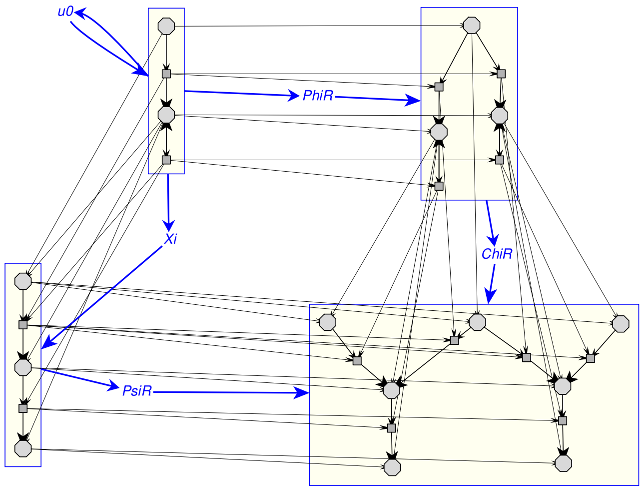
$$\mathsf{dom}\,(\Xi ; (\mathsf{ran}\,\Xi)^\sim) \sqsubseteq \mathsf{dom}\,(\mathsf{upa}\,\Phi)\ . \qquad \square$$

These two conditions could be unified into the following single inclusion:

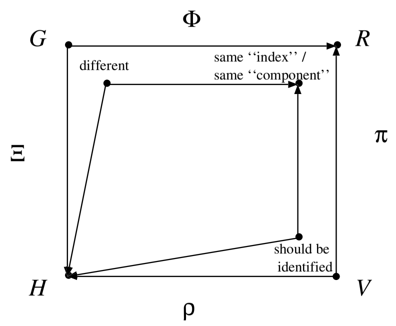
$$(\mathsf{injdom}\,\Xi)^\sim \sqcup \mathsf{dom}\,(\Xi ; (\mathsf{ran}\,\Xi)^\sim) \sqsubseteq \mathsf{dom}\,(\mathsf{upa}\,\Phi)$$

The first condition in particular ensures univalence of $\Phi$ on the pre-image of $u_2 \sqcap v_2$; the second condition extends univalence of $\Phi$ also to the pre-image of the last border of $\mathsf{ran}\,\Xi$ (or, equivalently because of interface preservation, of $u_2$).

In the following example (with empty interface $u_0$), $\mathsf{injdom}\,\Xi$ is empty, and $\Phi$ is univalent only on the top node, so ($\mathsf{spurious}$) does not hold, and the result construction we are going to define below delivers a result that does not even commute:




In order to make (spurious) superfluous, we would need some way to find out whether different graph parts in $R$ lie in the same "replication component" of replication via $\Phi$.




Since the identification of "replication components" would be similarly non-constructive as the equivalence relation complements needed for pullback complements, we rather keep the condition (spurious).

For obtaining a result, we use the intuition established above to merge aspects of the relational characterisations of pushouts (Def. 5.3.2) and pullbacks (Def. 5.1.2) into the amalgamated "weak pullout" construction:

**Definition 6.5.5** Given a gluing setup $(G_0, u_0, v_0, \Xi, \Phi)$, then a *weak pullout* for this setup is a cospan of morphisms $G_2 \xrightarrow{\Psi} G_3 \xleftarrow{\mathrm{X}} G_1$ that completes the given span to a *weak pullout diagram along* $u_0$ *(over* $v_0$*)*, which is a square



$$\begin{tikzcd} G_0 \arrow[r, "\Phi"] \arrow[d, "\Xi"'] & G_1 \arrow[d, "\mathrm{X}"] \\ G_2 \arrow[r, "\Psi"] & G_3 \end{tikzcd}$$

where the conditions below hold.

First we define two additional abbreviations:

$$v_3 \ := \ \mathsf{ran}\,(v_1 ; \mathrm{X}) \qquad\qquad w_2 \ := \ v_2 \sqcup (\mathsf{ran}\,\Xi)^{\sim}$$

In addition, let $f_0 := \mathsf{dom}\,(\mathsf{upa}\,\Phi)$ denote the domain of the univalent part of $\Phi$. Finally, we introduce the following abbreviations:

$$\begin{aligned} \Omega &:= \Phi^{\smile} ; u_0 ; f_0 ; \Xi \\ \Omega_f &:= \Phi^{\smile} ; f_0 ; \Xi \\ \Omega_u &:= \Phi^{\smile} ; u_0 ; \Xi \\ \Omega_v &:= \Phi^{\smile} ; v_0 ; \Xi \end{aligned}$$

For a weak pullout diagram, we demand that the following conditions hold:

– commutativity: $\Phi ; \mathrm{X} = \Xi ; \Psi$
– the *combination property*: $\mathbb{I} = \mathrm{X}^{\smile} ; c_1 ; \mathrm{X} \sqcup \Psi^{\smile} ; q_2 ; \Psi \sqcup (\Psi^{\smile} ; v_2 ; \Psi \sqcap \mathrm{X}^{\smile} ; v_1 ; \mathrm{X})$
– *alternative commutativity*: $\mathrm{X} ; \Psi^{\smile} = \Omega^{\rhd} ; \Phi^{\smile} ; \Xi$
– *semi-injectivity of* X: $\mathrm{X} ; \mathrm{X}^{\smile} = c_1 \sqcup \mathsf{dom}\,\Omega_v \sqcup \mathrm{X} ; \Psi^{\smile} ; \Omega_f^{\smile}$
– *semi-injectivity of* $\Psi$: $\Psi ; \Psi^{\smile} = c_2 \sqcup \mathsf{ran}\,\Omega_v \sqcup \Omega_u^{\smile} ; \mathrm{X} ; \Psi^{\smile}$ □

The combination property implies that X is univalent on $c_1$ and $\Psi$ is univalent on $q_2$.

The semi-injectivity conditions imply that $\Psi$ is almost-injective besides $u_2$ and, together with (spurious), that X is almost-injective besides $\mathsf{ran}\,(\mathsf{upa}\,\Phi)$.

For reasonable gluing setups, the weak pullout characterisation is monomorphic:

**Theorem 6.5.6** [←186] Given a weak pullout diagram for a reasonable gluing setup, then for every weak pullout diagram for the same setup there is a bijective mapping factoring it, if the Dedekind category underlying the discussion may be embedded in a Dedekind category with sharp products. □

The very technical proof is relegated to Sect. B.3. The indirect availability of sharp products is necessary only for establishing the parameter part of the combination property (Lemma B.3.3 on page 185), essentially because X is not even almost-injective on $v_1$. In fact, we do not need the availability of all products, but those products we need have to be sharp.

The weak pullout construction itself now may proceed along the intuition established with the figure on page 145. We make this construction precise in the following definition, which is also intended to serve as a guideline for implementations. Therefore, we implement a further optimisation: that part of the host parameter instantiation that is only in the image of the univalent part of $\Phi$ will not be replicated, so we remove it from the factor for the tabulation and include it in the directly copied part.

**Definition 6.5.7** [←150, 190] Let a reasonable gluing setup $(G_0, u_0, v_0, \Xi, \Phi)$ be given, consisting of a gluing object $G_0$ along $u_0$ over $v_0$, a host morphism $\Xi : G_0 \to G_2$, and an rhs-morphism $\Phi : G_0 \to G_1$.

Then the following defines the *direct result construction* for this setting as a cospan $G_1 \xrightarrow{X} G_3 \xleftarrow{\Psi} G_2$.

We continue to use the abbreviations from Def. 6.1.4 and Def. 6.5.5, and we introduce a few additional abbreviations for further important partial identities:

- $y_0 := \mathsf{injdom}\,\Xi$ is the domain of injectivity of $\Xi$; according to (spurious), it is possible that $\Phi$ is not univalent on $y_0$.
- $f_2 := \mathbb{I}_2 \sqcap \Xi\backslash(f_0;\mathbb{T}_{0,2})$ is a partial identity describing the maximal part of $G_2$ for which the pre-image wrt. $\Xi$ lies completely in $f_0 = \mathsf{dom}\,(\mathsf{upa}\,\Phi)$.
- $k_2 := h_2 \sqcup (v_2 \sqcap f_2) \sqcup b_2$ is that part of $G_2$ that will be directly copied into the result, consisting of the context and that part of the parameter instantiation that, because of univalence of $\Phi$, will not be replicated.

  Including the border $b_2$ between $u_2$ and $v_2$ is not strictly necessary — it will be identified with its other instances in the quotient construction below — but it considerably facilitates proofs.

For the parameter parts, we construct a direct tabulation $G_1 \xleftarrow{\pi} V_3 \xrightarrow{\rho} G_2$ for

$$W := \Phi^{\smile};v_0;\Xi;(v_2 \setminus f_2)$$

Then $G_4$ is the three-part direct sum via injections characterised as follows:

$$\begin{array}{llll}
\iota : G_1 \rightarrowtail\!\!\!\!\rightarrow G_4 & \iota;\iota^{\smile} = c_1 & \iota;\kappa^{\smile} = \perp\!\!\!\perp & \\
\kappa : V_3 \rightarrowtail G_4 & \kappa;\kappa^{\smile} = \mathbb{I} & \kappa;\lambda^{\smile} = \perp\!\!\!\perp & \iota^{\smile};\iota \sqcup \kappa^{\smile};\kappa \sqcup \lambda^{\smile};\lambda = \mathbb{I} \\
\lambda : G_2 \rightarrowtail\!\!\!\!\rightarrow G_4 & \lambda;\lambda^{\smile} = k_2 & \lambda;\iota^{\smile} = \perp\!\!\!\perp &
\end{array}$$

(It would be possible to use subobjects and standard direct sums for the same purpose, but this single definition allows to work with fewer relations and laws and thus shortens proofs.)

$$\begin{tikzcd}
G_0 \arrow[r, "\Phi"] \arrow[d, "\Xi"'] & G_1 \arrow[ddr, "\iota"] & \\
G_2 \arrow[drr, "\lambda"'] & V_3 \arrow[l, "\rho"'] \arrow[u, "\pi"] \arrow[dr, "\kappa"] & \\
& & G_4 \arrow[dr, "\theta"] & \\
& & & G_3
\end{tikzcd}$$



At this stage, we may define $X_0 : G_1 \to G_4$ and $\Psi_0 : G_2 \to G_4$ to obtain a first square completion of the starting diagram; this does, however, not yet commute:

$$\begin{aligned}
X_0 &:= \iota \sqcup \pi^{\smile};\kappa \sqcup \Phi^{\smile};\Xi;\lambda \\
\Psi_0 &:= \Xi^{\smile};\Phi;\iota \sqcup \rho^{\smile};\kappa \sqcup \lambda
\end{aligned}$$

Note that we have $\Psi_0 = \Xi^{\smile};u_0;\Phi;\iota \sqcup \rho^{\smile};\kappa \sqcup \lambda$ because of Def. 6.1.2.ii). For making the result commute, we now have to identify borders between the different components of the three-part sum in $G_4$. For this purpose we first define separate relations each relating one pair of borders:

$$\begin{aligned}
Q_1 &:= \iota^{\smile};\pi^{\smile};\kappa \\
Q_2 &:= \lambda^{\smile};\rho^{\smile};\kappa \\
Q_3 &:= \iota^{\smile};\Phi^{\smile};\Xi;\lambda \\
Q_4 &:= \iota^{\smile};\Phi^{\smile};y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};\Phi;\iota
\end{aligned}$$

The following drawing documents the typing of the essential parts of these $Q_i$, leaving out the leading and trailing injections for the sum in $G_4$.

$$\begin{tikzcd}
V_3 & c_1 \arrow[l, "Q_1"'] \arrow[dl, "Q_3"] \arrow[loop right, "Q_4"] \\
k_2 \arrow[u, "Q_2"] &
\end{tikzcd}$$

$Q_4$ transfers interface identifications from $u_0;\Xi$ to $c_1;X$. Without the $y_0^{\sim}$, it would overly destroy injectivity inside $u_1$. Because of (spurious), we have $y_0^{\sim} \sqsubseteq f_0$, so the effect of $Q_4$ is limited to the image of the univalent part of $\Phi$.

Since $\Phi$ is almost-injective on $v_0$, no corresponding identification relation needs to be established for $v_2$.

Now we define $\Theta : G_4 \leftrightarrow G_4$ as the equivalence relation connecting the border components that are replicated in different parts of the sum:

$$\Theta := (Q \sqcup Q^{\smile})^* \;, \qquad \text{where} \qquad Q := Q_1 \sqcup Q_2 \sqcup Q_3 \sqcup Q_4 \;.$$

Finally, $G_3$ is the quotient of $G_4$ by $\Theta$, with total, univalent and surjective projection $\theta : G_4 \twoheadrightarrow G_3$, and we define

$$X := X_0;\theta \qquad \text{and} \qquad \Psi := \Psi_0;\theta \;. \qquad \square$$

Even though there are relatively many components to take care of, the reader will notice that the "gluing together" is essentially straightforward and strictly oriented at the guidelines lined out along the drawing on page 145.

The proof of correctness is therefore not very creative, but rather lengthy because of the many interconnections involved; it has been relegated to Sect. B.4, starting on page 190. It proves the following:

**Theorem 6.5.8** If for a reasonable gluing setup the tabulation, sub-direct sum, and quotient of Def. 6.5.7 exist, then this direct result construction produces a weak pullout. □

For left-linear rules, we may again use the straight host construction for obtaining a weak pullout complement. The preconditions are slightly changed, again: In the following, the difference to the preconditions of Theorem 6.3.1 is in (ii), where X need not be almost-injective on $v_1$ anymore:

**Theorem 6.5.9** [←199] Let a gluing object $\mathcal{G}$ along $u_0$ over $v_0$ and a standard gluing morphism $\Phi : \mathcal{G} \to \mathcal{L}$ be given, and a morphism $\mathrm{X} : \mathcal{L} \to \mathcal{A}$. Define:

$$
\begin{array}{lllllllll}
v_1 & := & \mathsf{ran}\,(v_0;\Phi) & u_1 & := & \mathsf{ran}\,(u_0;\Phi) & c_1 & := & v_1^{\smile} \sqcup u_1 \\
v_3 & := & \mathsf{ran}\,(v_1;\mathrm{X}) & u_3 & := & \mathsf{ran}\,(u_1;\mathrm{X}) & c_3 & := & (\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup u_3
\end{array}
$$

If the following conditions are satisfied:

(i) $c_1;\mathrm{X}$ is univalent, and $c_1 \sqsubseteq \mathsf{dom}\,\mathrm{X}$,
(ii) X is almost-injective besides $\mathsf{ran}\,\Phi$,
(iii) $\mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\sim} \sqsubseteq u_1;\mathrm{X}$,
(iv) $v_3 \sqcap v_3^{\sim} \sqsubseteq u_3$ and
(v) $\mathrm{X};u_3 \sqsubseteq u_1;\mathrm{X}$.

then the straight host construction of Def. 5.4.6 delivers a weak pullout complement. □

The proof of this theorem may be found in Sect. B.5.

For a *double-weak-pullout approach*, these last two theorems together establish the technicalities, limited to left-linear rules. However, this double-weak-pullout approach allows and preserves overlap between parameters.

## Chapter 7

# Conclusion and Outlook

In Chapters 2–4 we showed how unary signatures naturally give rise to strict Dedekind categories of graph structures. This means that a concept of "relational graph-structure homomorphism" makes sense, and allows the full range of relation-algebraic reasoning as far as complements are not involved.

We then limited ourselves to an abstract setting of strict Dedekind categories. In Chapter 5 we investigated the main categorical approaches to graph rewriting, namely the double- and single-pushout approach, and the single- and double-pullback approach, and in summary, the relation-algebraic approach turns out to be a good middle road for formalising categoric graph-structure transformation:

- Much of the literature of the categoric approaches to graph rewriting uses a "simplistic category-theoretic setting": pushouts, pullbacks, monomorphisms, and other concepts on that level are used, and contribute to attaining a useful level of abstraction, while still being accessible from a reasonable level of prerequisites.

  However, not all important concepts can be formalised on this level: that part of the literature then resorts to component-wise definitions of, for example, the gluing condition or conflict-freeness.

- It is well-known, and frequently pointed out in the literature, that graph structures with conventional graph-structure homomorphisms give rise to topoi. However, not many of those interested in applications of graph transformations are equipped with the prerequisites for being able to follow topos-theoretic arguments. Therefore, most of the literature abstains from actually using the power of graph-structure topoi.

- The Dedekind category approach put forward in this thesis, although formally embedded in category theory, in fact offers itself more to a typical *relational* style of reasoning than to the typical category-theoretic style.

  This kind of relational reasoning, with its inherent vicinity to matrix calculations and linear algebra, is probably much more accessible to many people with diverse backgrounds and an interest in graph transformation than even the "simplistic category-theoretic setting". Although in comparison with the latter, Dedekind categories come equipped with more operations and laws, many of these operations and laws are actually more familiar to prospective users of graph transformation than even pushouts or pullbacks.

  In addition, as we have shown, (strict) Dedekind categories allow to capture all the central concepts of the categoric approaches to graph transformation in a component-free manner.

Furthermore, the relation-algebraic approach allows rather fine-grained customisation and combination of different approaches, as we documented with the pullout approaches of

Chapter 6, which served as a “natural” incorporation of variable concepts into the double-pushout approach, achieved via an amalgamation of pushouts and pullbacks, and made possible through the unifying view of relational graph-structure homomorphisms.

For reasons of space we omitted discussion of labelled structures — in the unary algebra approach, there are two possibilities to formalise labelled structures:

- Some sorts are designated as “label sorts”, and only those morphisms are considered that for all label sorts have identities as the corresponding components.
- A special “label algebra” (similar to the alphabet graph on page 20) is selected, and mappings from other algebras to this label algebra are considered as objects, and relational morphisms respecting these label mappings as morphisms. This corresponds to the comma-category approach that is also used under the name “typed graphs”, see for example [RB88, BC99].

In both cases it is easy to establish that labelled unary algebras again give rise to Dedekind categories, and that the standard construction of subobjects, quotients, direct sums and direct products remain possible.

Some more work will need to deal with partial labellings and interactions between labellings and parameter parts.

Future work will also have to tackle the usual body of questions about interaction of and relations between several direct derivation steps, so that parallel and sequential independence conditions, and embedding, amalgamation and distribution conditions may be established, see [CMR+97, Sect. 3.2].

In addition, it will be interesting to investigate categories of hierarchical graphs, and how these may be incorporated into our approach in such a way that parameters can be matched across different levels of the hierarchy.

Also, one may want to investigate applications of Dedekind categories of graph structures over bases that are not just the Dedekind category *Rel* of concrete relations, but perhaps fuzzy relations, or other Dedekind categories of abstract or concrete graph structures.

# Bibliography

[ABH+92] Chritiene Aarts, Roland C. Backhouse, Paul Hoogendijk, Ed Voermans, and Jaap van der Woude. A relational theory of datatypes. Working document, December 1992. 387 pp., available at http://www.cs.nott.ac.uk/~rcb/MPC/book.ps.gz.

[Ban93] Richard Banach. A fibration semantics for extended term graph rewriting. In Sleep et al. [SPvE93], chapter 7, pages 91–100.

[Ban94] Richard Banach. Term graph rewriting and garbage collection using opfibrations. *Theoretical Computer Science*, 131(1):29–94, August 1994.

[Bar79] M. Barr. *∗-Autonomous Categories*, volume 752 of *Lect. Notes in Math.* Springer, 1979.

[Bau95] Michel Bauderon. Parallel rewriting through the pullback approach. *Electronic Notes in Computer Science*, 2, 1995. SEGRAGRA ’95.

[Bau97] Michel Bauderon. A uniform approach to graph rewriting: The pullback approach. In Manfred Nagl, editor, *Graph Theoretic Concepts in Computer Science, WG ’96*, volume 1017 of *LNCS*, pages 101–115. Springer, 1997.

[BC99] Richard Banach and Andrea Corradini. Abstract diagrams and an opfibration account of typed graph transformations, 1999. submitted to TCS.

[BdM97] Richard S. Bird and Oege de Moor. *Algebra of Programming*, volume 100 of *International Series in Computer Science*. Prentice Hall, 1997.

[BFH85] Paul Boehm, Harald-Reto Fonio, and Annegret Habel. Amalgamation of graph transformations with applications to synchronization. In H. Ehrig, C. Floyd, M. Nivat, and J. Thatcher, editors, *Mathematical Foundations of Software Development*, volume 185 of *LNCS*, pages 267–283. Springer, 1985.

[BHSV94] Rudolf Berghammer, Armando M. Haeberer, Gunther Schmidt, and Paulo A. S. Veloso. Comparing two different approaches to products in abstract relation algebra. In Nivat et al. [NRRS94], pages 167–176.

[BJ96] Michel Bauderon and Hélène Jacquet. Categorical product as a generic graph rewriting mechanism. Technical Report 1166–97, LaBRI, University of Bordeaux, 1996. see also [BJ01].

[BJ97] Michel Bauderon and Hélène Jacquet. Node rewriting in hypergraphs. In Fabrizio d’Amore, Paola G. Franciosa, and Alberto Marchetti-Spaccamela, editors, *Graph Theoretic Concepts in Computer Science, WG ’96*, volume 1197 of *LNCS*, pages 31–43. Springer, 1997.

[BJ01] Michel Bauderon and Hélène Jacquet. Pullback as a generic graph rewriting mechanism. *Applied Categorical Structures*, 9(1):65–82, 2001.

[BKS97] Chris Brink, Wolfram Kahl, and Gunther Schmidt, editors. *Relational Methods in Computer Science*. Advances in Computing Science. Springer, Wien, New York, 1997.

[BMST99] R. Bardohl, M. Minas, A. Schuerr, and G. Taentzer. Application of graph transformation to visual languages. In Ehrig et al. [EEKR99], chapter 3, pages 105–180.

[Boo47] George Boole. *The Mathematical Analysis of Logic, Being an Essay Toward a Calculus of Deductive Reasoning*. Macmillan, Cambridge, 1847.

[BRTV96] P. Burmeister, F. Rosselló, J. Torrens, and G. Valiente. Algebraic transformation of partial algebras I: Double-pushout approach, February 1996.

[BS99] E. Barendsen and S. Smetsers. Graph rewriting aspects of functional programming. In Ehrig et al. [EEKR99], chapter 2, pages 63–102.

[BSZ86] Rudolf Berghammer, Gunther Schmidt, and Hans Zierer. Symmetric quotients. Technical Report TUM-INFO 8620, Technische Universität München, Fakultät für Informatik, 1986. 18 p.

[BSZ89] Rudolf Berghammer, Gunther Schmidt, and Hans Zierer. Symmetric quotients and domain constructions. *Inform. Process. Lett.*, 33(3):163–168, 1989.

[Bur86] Peter Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras (Introduction to Theory and Application of Partial Algebras — Part I)*, volume 32 of *Mathematical Research*. Akademie-Verlag Berlin, 1986.

[But98] Carsten Butz. Finitely presented Heyting algebras. http://euclid.math.mcgill.ca/butz/publications/abstracts.html#heyting, October 1998.

[BW84] Michael Barr and Charles Wells. *Toposes, Triples and Theories*, volume 278 of *Grundlehren Math. Wiss.* Springer, Berlin, 1984.

[BW90] Michael Barr and Charles Wells. *Category Theory for Computing Science*. Prentice Hall International Series in Computer Science. Prentice Hall, 1990.

[BZ86] Rudolf Berghammer and Hans Zierer. Relational algebraic semantics of deterministic and nondeterministic programs. *Theoretical Computer Science*, 43:123–147, 1986.

[C+96] J. Cuny et al., editors. *Graph-Grammars and Their Application to Computer Science, 5th International Workshop, Williamsburg, VA, USA, November 1994, Selected Papers*, volume 1073 of *LNCS*. Springer, November 1996.

[Car82] Rodrigo Cardoso. *Untersuchung paralleler Programme mit relationenalgebraischen Methoden*. Diplomarbeit under supervision of Gunther Schmidt, TU München, 1982.





[CER78] Volker Claus, Hartmut Ehrig, and Grzegorz Rozenberg, editors. *Graph-Grammars and Their Application to Computer Science and Biology, International Workshop*, volume 73 of *LNCS*. Springer, November 1978.

[CMR+97] Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, Reiko Heckel, and Michael Löwe. Algebraic approaches to graph transformation, part I: Basic concepts and double pushout approach. In Rozenberg [Roz97], chapter 3, pages 163–245.

[CR93a] Andrea Corradini and Francesca Rossi. Hyperedge replacement jungle rewriting for term-rewriting systems and logic programming. In Courcelle and Rozenberg [CR93b], pages 7–48. Theoretical Computer Science **109** (1–2).

[CR93b] B. Courcelle and G. Rozenberg, editors. *Selected Papers of the International Workshop on Computing by Graph Transformation, Bordeaux, France, March 21–23, 1991*. Elsevier, 1993. Theoretical Computer Science **109** (1–2).

[CW87] Aurelio Carboni and Robert F. C. Walters. Cartesian bicategories I. *Annals of Pure and Applied Algebra*, 49:11–32, 1987.

[Des99] Jules Desharnais. Monomorphic characterization of $n$-ary direct products. *Information Sciences*, 119(3–4):275–288, December 1999.

[Des00] Jules Desharnais, editor. *Fifth International Seminar on Relational Methods in Computer Science, 10–14 January 2000, Valcartier, Québec, Canada, Participant's Proceedings*. Université Laval, Département d'informatique, 2000.

[DG00] Dan Dougherty and Claudio Gutiérrez. Normal forms and reduction for theories of binary relations. In Leo Bachmair, editor, *Rewriting Techniques and Applications, Proc. RTA 2000*, volume 1833 of *LNCS*, pages 95–109. Springer, 2000.

[DHP00a] F. Drewes, B. Hoffmann, and Detlef Plump. Hierarchical graph transformation. In Jerzy Tiuryn, editor, *FoSSaCS 2000*, volume 1784 of *LNCS*, pages 98–113. Springer, 2000.

[DHP00b] Frank Drewes, Berthold Hoffmann, and Detlef Plump. Hierarchical graph transformation. Technical Report 1/00, Fachbereich Mathematik / Informatik, Universität Bremen, 2000.

[DKH97] Frank Drewes, Hans-Jörg Kreowski, and Annegret Habel. Hyperedge replacement graph grammars. In Rozenberg [Roz97], chapter 2, pages 95–162.

[DM60] Augustus De Morgan. On the Syllogism: IV; and on the Logic of Relations. *Trans. of the Cambridge Philosophical Society*, 10:331–358, 1860. (dated 12 November 1859) Reprinted in [DM66].

[DM66] Augustus De Morgan. *On the Syllogism, and Other Logical Writings*. Yale Univ. Press, New Haven, 1966.

[DMN97] Jules Desharnais, Ali Mili, and Thanh Tung Nguyen. Refinement and demonic semantics. In Brink et al. [BKS97], chapter 11, pages 166–183.

[dRE98] Willem-Paul de Roever and Kai Engelhardt. *Data Refinement: Model-Oriented Proof Methods and their Comparison*, volume 47 of *Cambridge Tracts Theoret. Comput. Sci.* Cambridge Univ. Press, 1998.

[DvGB97] Henk Doornbos, Netty van Gasteren, and Roland Backhouse. Programs and datatypes. In Brink et al. [BKS97], chapter 10, pages 150–165.

[EB94] Hartmut Ehrig and Roswitha Bardohl. Specification techniques using dynamic abstract data types and application to shipping software. In *Proc. Shanghai Workshop on Software Technology*, pages 70–85, 1994.

[EEKR99] Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors. *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 2: Applications, Languages and Tools.* World Scientific, Singapore, 1999.

[EEKR00] Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors. *Theory and Application of Graph Transformations, 6th International Workshop, TAGT'98, Paderborn, Germany, November 1998, Selected Papers*, volume 1764 of *LNCS*. Springer, 2000.

[EHK$^+$97] Hartmut Ehrig, Reiko Heckel, Martin Korff, Michael Löwe, Leila Ribeiro, Annika Wagner, and Andrea Corradini. Algebraic approaches to graph transformation, part II: Single pushout approach and comparison with double pushout approach. In Rozenberg [Roz97], chapter 4, pages 247–312.

[EHKPP91] Hartmut Ehrig, Annegret Habel, Hans-Jörg Kreowski, and Francesco Parisi-Presicce. From graph grammars to high level replacement systems. In Ehrig et al. [EKR91], pages 269–287.

[EHL$^+$00] Hartmut Ehrig, Reiko Heckel, Mercè Llabreés, Fernando Orejas, Julia Padberg, and Grzegorz Rozenberg. Double-pullback graph transitions: A rule-based framework with incomplete information. In Ehrig et al. [EEKR00], pages 85–102.

[Ehr78] Hartmut Ehrig. Introduction to the algebraic theory of graph grammars. In Claus et al. [CER78], pages 1–69.

[Ehr87] Hartmut Ehrig. Tutorial Introduction to the Algebraic Approach of Graph-Grammars. In Ehrig et al. [ENRR87], pages 3–14.

[EHTE97] Gregor Engels, Reiko Heckel, Gabriele Taentzer, and Hartmut Ehrig. A combined reference model- and view-based approach to system specification. *Intl. Journal of Software and Knowledge Engineering*, 7(4):457–477, 1997.

[EKL91] Hartmut Ehrig, Martin Korff, and Michael Löwe. Tutorial introduction to the algebraic approach of graph grammars based on double and single pushouts. In Ehrig et al. [EKR91], pages 24–37.



[EKR91] Hartmut Ehrig, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors. *Graph-Grammars and Their Application to Computer Science, 4th International Workshop*, volume 532 of *LNCS*. Springer, March 1991.

[EL93] Hartmut Ehrig and Martin Löwe. The ESPRIT Basic Research Working Group COMPUGRAPH "Computing by Graph Transformation": a survey. In Courcelle and Rozenberg [CR93b], pages 3–6. Theoretical Computer Science **109** (1–2).

[EM85] Hartmut Ehrig and Bernd Mahr. *Fundamentals of Algebraic Specification (Volumes 1 and 2).* Springer, 1985.

[ENR82] Hartmut Ehrig, Manfred Nagl, and Grzegorz Rozenberg, editors. *Graph-Grammars and Their Application to Computer Science, 2nd International Workshop*, volume 153 of *LNCS*. Springer, 1982.

[ENRR87] Hartmut Ehrig, Manfred Nagl, Grzegorz Rozenberg, and Azriel Rosenfeld, editors. *Graph-Grammars and Their Application to Computer Science, 3rd International Workshop*, volume 291 of *LNCS*. Springer, December 1987.

[EPP94] Hartmut Ehrig and Francesco Parisi-Presicce. Interaction between algebraic specification grammars and modular system design. In Nivat et al. [NRRS94], pages 217–224.

[EPS73] Hartmut Ehrig, M. Pfender, and H. J. Schneider. Graph grammars: An algebraic approach. In *Proc. IEEE Conf. on Automata and Switching Theory, SWAT '73*, pages 167–180, 1973.

[ER97] Joost Engelfriet and Grzegorz Rozenberg. Node replacement graph grammars. In Rozenberg [Roz97], chapter 1, pages 1–94.

[ET00] Hartmut Ehrig and Gabriele Taentzer, editors. *Proc. Joint APPLIGRAPH / GETGRATS Workshop on Graph Transformation Systems (GRATRA 2000).* TU Berlin, Fachbereich 13: Informatik, 2000. Report Nr. 2000-2.

[FK87] Joseph H. Fasel and Robert M. Keller, editors. *Graph Reduction: Proceedings of a Workshop*, volume 279 of *LNCS*, Santa Fé, New Mexico, 1987. Springer.

[FK98] Hitoshi Furusawa and Wolfram Kahl. A study on symmetric quotients. Technical Report 1998-06, Fakultät für Informatik, Universität der Bundeswehr München, December 1998.

[FS90] Peter J. Freyd and Andre Scedrov. *Categories, Allegories*, volume 39 of *North-Holland Mathematical Library.* North-Holland, Amsterdam, 1990.

[Fur98] Hitoshi Furusawa. *Algebraic Formalisations of Fuzzy Relations and Their Representation Theorems.* PhD thesis, Department of Informatics, Kyushu University, March 1998.

[FW77] M. Farzan and D.A. Waller. Kronecker products and local joins of graphs. *Can. J. Math.*, XXIX(2):255–269, 1977.

[GKOT00] Yuri Gurevich, Philipp W. Kutter, Martin Odersky, and Lothar Thiele, editors. *Abstract State Machines, Theory and Applications, International Workshop ASM 2000, Monte Veritá, Switzerland*, volume 1912 of *LNCS*. Springer, 2000.

[Gol84] Robert Goldblatt. *Topoi, The Categorical Analysis of Logic*, volume 98 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, revised edition, 1984.

[Grä78] George Grätzer. *General Lattice Theory.* Birkhäuser, Basel, 1978.

[Grä79] George Grätzer. *Universal Algebra.* Springer, New York, second edition edition, 1979. (first edition 1968).

[Gur91] Yuri Gurevich. Evolving algebras: A tutorial introduction. *Bull. of the European Association for Theoretical Computer Science (EATCS)*, 43:264–286, 1991.

[Gut99] Claudio Gutiérrez. *The Arithmetic and Geometry of Allegories: Normal Forms and Complexity of a Fragment of the Theory of Relations.* PhD thesis, Wesleyan University, 1999.

[Hec98] Reiko Heckel. *Open Graph Transformation Systems, A New Approach to the Compositional Modelling of Concurrent and Reactive Systems.* PhD thesis, Fachbereich 13 — Informatik der Technischen Universität Berlin, 1998.

[HEWC97] Reiko Heckel, Hartmut Ehrig, Uwe Wolter, and Andrea Corradini. Loose semantics and constraints for graph transformation systems. Technical Report 97/7, Fachbereich Informatik, TU Berlin, 1997.

[HF98] Armando Haeberer and Marcelo Frias. Relational methods in computer science, special issue devoted to RelMiCS 2 in Paraty, July 1995. *Journal of the IGPL*, 6(2), March 1998.

[HFBV97] Armando Haeberer, Marcelo Frias, Gabriel Baum, and Paulo Veloso. Fork algebras. In Brink et al. [BKS97], chapter 4, pages 54–69.

[HMP00] Annegret Habel, Jürgen Müller, and Detlef Plump. Double-pushout graph transformation revisited. Technical report, Fachbereich Mathematik und Informatik, Universität Bremen, 2000.

[HMP01] Annegret Habel, Jürgen Müller, and Detlef Plump. Double-pushout graph transformation revisited. *Mathematical Structures in Computer Science*, 11(4), 2001.

[HP91] B. Hoffmann and Detlef Plump. Implementing term rewriting by jungle evaluation. *Informatique théorique et applications/Theoretical Informatics and Applications*, 25(5):445–472, 1991.

[Jac99] Hélène Jacquet. *Une approche catégorique de la réécriture de sommets dans les graphes.* PhD thesis, Université Bordeaux 1, January 1999.





[JKH00] Hélène Jacquet and Renate Klempien-Hinrichs. Node replacement in hypergraphs: Translating NCE rewriting into the pullback approach. In Ehrig et al. [EEKR00], pages 117–130.

[Jón72] Bjarni Jónsson. *Topics in Universal Algebra*, volume 250 of *Lect. Notes in Math.* Springer, 1972.

[Jon90] C. V. Jones. An introduction to graph-based modeling systems, part I: Overview. *ORSA Journal on Computing*, 2(2):136–151, 1990.

[Jon91] C. V. Jones. An introduction to graph-based modeling systems, part II: Graph-grammars and the implementation. *ORSA Journal on Computing*, 3(3):180–206, 1991.

[Jon92] C. V. Jones. Attributed graphs, graph-grammars, and structured modeling. *Annals of OR*, 38:281–324, 1992.

[JS67] J. Johnson and R. L. Seifert. A survey of multi-unary algebras. Mimeographed seminar notes, U. C. Berkeley, 1967. 26 pages.

[JS99] Ali Jaoua and Gunther Schmidt. Relational methods in computer science, special issue devoted to RelMiCS 3 in Hammamet, January 1997. *Information Sciences*, 119(3–4):131–314, December 1999.

[Kah96] Wolfram Kahl. *Algebraische Termgraphersetzung mit gebundenen Variablen.* Reihe Informatik. Herbert Utz Verlag Wissenschaft, München, 1996. ISBN 3-931327-60-4; also Doctoral Diss. at Univ. der Bundeswehr München, Fakultät für Informatik.

[Kah97] Wolfram Kahl. A fibred approach to rewriting — how the duality between adding and deleting cooperates with the difference between matching and rewriting. Technical Report 9702, Fakultät für Informatik, Universität der Bundeswehr München, May 1997.

[Kah99] Wolfram Kahl. Relational matching for graphical calculi of relations. *Information Sciences*, 119(3–4):253–273, December 1999.

[Kaw73a] Yasuo Kawahara. Matrix calculus in I-categories and an axiomatic characterization of relations in a regular category. *Mem. Fac. Sci. Kyushu Univ. Ser. A*, 27(2):249–273, 1973.

[Kaw73b] Yasuo Kawahara. Notes on the universality of relational functors. *Mem. Fac. Sci. Kyushu Univ. Ser. A*, 27(2):275–289, 1973.

[Kaw73c] Yasuo Kawahara. Relations in categories with pullbacks. *Mem. Fac. Sci. Kyushu Univ. Ser. A*, 27(1):149–173, 1973.

[Kaw90] Yasuo Kawahara. Pushout-complements and basic concepts of grammars in toposes. *Theoretical Computer Science*, 77:267–289, 1990.

[Ken87] Richard Kennaway. On "On Graph Rewritings". *Theoretical Computer Science*, 52:37–58, 1987.

[Ken91] Richard Kennaway. Graph rewriting in some categories of partial morphisms. In Ehrig et al. [EKR91], pages 490–504.

[KFM99] Yasuo Kawahara, Hitoshi Furusawa, and Masao Mori. Categorical representation theorems of fuzzy relations. *Information Sciences*, 119(3–4):235–251, December 1999.

[Kre87] Hans-Jörg Kreowski. Is Parallelism Already Concurrency? Part 1: Derivations in Graph Grammars. In Ehrig et al. [ENRR87], pages 343–360.

[KS00] Wolfram Kahl and Gunther Schmidt. Exploring (finite) Relation Algebras using Tools written in Haskell. Technical Report 2000-02, Fakultät für Informatik, Universität der Bundeswehr München, October 2000. http://ist.unibw-muenchen.de/relmics/tools/RATH/.

[KSW94] Peter Kempf, Gunther Schmidt, and Michael Winter. Konstruktion semantischer Bereiche aus algebraischen Spezifikationen. Technical Report 94/04, Univ. der Bundeswehr München, Fakultät für Informatik, 1994.

[KW87] Hans-Jörg Kreowski and Anne Wilharm. Is Parallelism Already Concurrency? Part 2: Non Sequential Processes in Graph Grammars. In Ehrig et al. [ENRR87], pages 361–377.

[Law63] F. William Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci. USA*, 50:869–872, 1963.

[LC79] Aristid Lindenmayer and Karel Culik II. Growing cellular systems: Generation of graphs by parallel rewriting. *Int. J. General Systems*, 5:45–55, 1979.

[LE91] Michael Löwe and Hartmut Ehrig. Algebraic approach to graph transformation based on single pushout derivations. In R. H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science, WG '90*, volume 484 of *LNCS*, pages 338–353. Springer, 1991.

[Löw90] Michael Löwe. Algebraic approach to graph transformation based on single pushout derivations. Technical Report 90/05, TU Berlin, 1990.

[Löw93] Martin Löwe. Algebraic approach to single-pushout graph transformation. In Courcelle and Rozenberg [CR93b], pages 181–224. Theoretical Computer Science **109** (1–2).

[Mad95] Roger Duncan Maddux. On the derivation of identities involving projection functions. In Csirmaz, Gabbay, and de Rijke, editors, *Logic Colloquium '92*, pages 145–163, Stanford, January 1995. Center for the Study of Language and Information Publications.





[NRRS94] Maurice Nivat, Charles Rattray, Teodore Rus, and Giuseppe Scollo, editors. *Proc. $3^{rd}$ Internat. Conf. Algebraic Methodology and Software Technology, Enschede, June 21–25*, Workshops in Computing. Springer, 1994.

[NSM00] Manfred Nagl, Andreas Schürr, and Manfred Münch, editors. *Applications of Graph Transformations with Industrial Relevance, Proc. AGTIVE'99, Kerkrade, The Netherlands, Spt. 1999*, volume 1779 of *LNCS*. Springer, 2000.

[Orl98] Ewa Orlowska, editor. *Relational Methods in Logic, Algebra and Computer Science, 4th International Seminar RelMiCS, Warsaw, Poland, 14–20 September 1998, Extended Abstracts.* Stefan Banach International Mathematical Center, Warsaw, 1998.

[OS80] Jean-Pierre Olivier and Dany Serrato. Catégories de Dedekind. Morphismes dans les catégories de Schröder. *C. R. Acad. Sci. Paris Ser. A-B*, 290:939–941, 1980.

[OS82] Jean-Pierre Olivier and Dany Serrato. Approach to an axiomatic study on the fuzzy relations on finite sets. In M. M. Gupta and E. Sanchez, editors, *Fuzzy Information and Decision Processes*. North-Holland, 1982.

[OS95] Jean-Pierre Olivier and Dany Serrato. Squares and rectangles in relation categories – three cases: Semilattice, distributive lattice and boolean non-unitary. *Fuzzy Sets and Systems*, 72:167–178, 1995.

[Pad82] Peter Padawitz. Graph grammars and operational semantics. *Theoretical Computer Science*, 19(1):117–141, 1982.

[Pei70] Charles Sanders Peirce. Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic. *Memoirs of the American Academy of Sciences*, 9:317–378, 1870. Reprint by Welch, Bigelow and Co., Cambridge, MA, 1870, pp. 1–62. Also reprinted in [Pei33] and [Pei84].

[Pei33] Charles Sanders Peirce. *C. S. Peirce Collected Papers*. Harvard Univ. Press, Cambridge, 1933. edited by C. Hartshorne and P. Weiss.

[Pei84] Charles Sanders Peirce. *Writings of Charles S. Peirce, A Chronological Edition*. Indiana Univ. Press, Bloomington, 1984. edited by Edward C. Moore, Max H. Fisch, Christian J. W. Kloesel, Don D. Roberts, and Lynn A. Ziegler.

[PH96] Detlef Plump and Annegret Habel. Graph unification and matching. In Cuny et al. [$C^+96$], pages 75–89.

[PL90] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, New York, 1990.

[Plu99] Detlef Plump. Term graph rewriting. In Ehrig et al. [EEKR99], chapter 1, pages 3–61.

[PP93] Francesco Parisi-Presicce. Single vs. double pushout derivation of graphs. In Ernst Mayr, editor, *Graph Theoretic Concepts in Computer Science, WG '92*, volume 657 of *LNCS*, pages 248–262. Springer, 1993.

[PR69] John L. Pfaltz and A. Rosenfeld. Web grammars. In *Proc. Int. Joint Conf. Art. Intelligence*, pages 609–619, Washington, 1969.

[PvE93] Rinus Plasmeijer and Marko van Eekelen. *Functional Programming and Parallel Graph Rewriting*. International Computer Science Series. Addison-Wesley, 1993.

[Rao84] J. C. Raoult. On graph rewritings. *Theoretical Computer Science*, 32:1–24, 1984.

[RB88] D.E. Rydeheard and R.M. Burstall. *Computational Category Theory*. Prentice Hall, 1988.

[Rig48] Jacques Riguet. Relations binaires, fermetures, correspondances de Galois. *Bull. Soc. Math. France*, 76:114–155, 1948.

[Roz97] Grzegorz Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 1: Foundations*. World Scientific, Singapore, 1997.

[Sch95] Ernst Schröder. *Vorlesungen über die Algebra der Logik (exacte Logik)*. Teubner, Leipzig, 1895. Vol. 3, Algebra und Logik der Relative, part I, $2^{nd}$ edition published by Chelsea, 1966.

[Sch70] H. J. Schneider. Chomsky-Systeme für partielle Ordnungen. Arbeitsber. d. Inst. f. Math. Masch. u. Datenver. 3, 3, Friedrich-Alexander-Universität Erlangen-Nürnberg, 1970.

[Sch76] Gunther Schmidt. Eine relationenalgebraische Auffassung der Graphentheorie. In H. Noltemeier, editor, *Graphen, Algorithmen, Datenstrukturen (WG '76)*, pages 315–325. Hanser, 1976.

[Sch77] Gunther Schmidt. Programme als partielle Graphen. Habil. Thesis, Fachbereich Mathematik der Technischen Univ. München, Bericht 7813, 1977. English as [Sch81a, Sch81b].

[Sch81a] Gunther Schmidt. Programs as partial graphs I: Flow equivalence and correctness. *Theoretical Computer Science*, 15:1–25, 1981.

[Sch81b] Gunther Schmidt. Programs as partial graphs II: Recursion. *Theoretical Computer Science*, 15(2):159–179, 1981.

[Sch90] Hans Jürgen Schneider. Describing distributed systems by categorical graph grammars. In Manfred Nagl, editor, *Graph-Theoretic Concepts in Computer Science, WG '89*, volume 411 of *LNCS*, pages 121–135. Springer, 1990.



[Sch97] Andy Schürr. Programmed graph replacement systems. In Rozenberg [Roz97], pages 479–546.

[SE93] Hans Jürgen Schneider and Hartmut Ehrig, editors. *Graph Transformations in Computer Science, Proc. International Workshop Dagstuhl Castle, Germany, January 1993*, volume 776 of *LNCS*. Springer-Verlag, 1993.

[SHW97] Gunther Schmidt, Claudia Hattensperger, and Michael Winter. Heterogeneous relation algebra. In Brink et al. [BKS97], chapter 3, pages 39–53.

[Spi89] J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall International Series in Computer Science. Prentice-Hall, 1989. Out of print; available via URL: http://spivey.oriel.ox.ac.uk/~mike/zrm/.

[SPvE93] M.R. Sleep, M.J. Plasmeijer, and M.C.J.D. van Eekelen, editors. *Term Graph Rewriting: Theory and Practice*. Wiley, 1993.

[SS85] Gunther Schmidt and Thomas Ströhlein. Relation algebras — concept of points and representability. *Discrete Math.*, 54:83–92, 1985.

[SS89] Gunther Schmidt and Thomas Ströhlein. *Relationen und Graphen*. Mathematik für Informatiker. Springer, Berlin, 1989. English as [SS93].

[SS93] Gunther Schmidt and Thomas Ströhlein. *Relations and Graphs, Discrete Mathematics for Computer Scientists*. EATCS-Monographs on Theoretical Computer Science. Springer, 1993.

[SWZ99] Andy Schürr, Andreas J. Winter, and A. Zündorf. The PROGRES approach: Language and environment. In Ehrig et al. [EEKR99], chapter 13, pages 487–550.

[Tar41] Alfred Tarski. On the calculus of relations. *Journal of Symbolic Logic*, 6(3):73–89, 1941.

[Tar52] Alfred Tarski. On representable relation algebras. *Bull. Amer. Math. Soc.*, 58:172, 1952. Abstract 145.

[TG87] Alfred Tarski and Steven Givant. *A Formalization of Set Theory without Variables*, volume 41 of *Amer. Math. Soc. Colloq. Publ.* Amer. Math. Soc., Providence, 1987.

[Vic89] Steven Vickers. *Topology via Logic*, volume 5 of *Cambridge Tracts Theoret. Comput. Sci.* Cambridge Univ. Press, 1989.

[WB98] MichałWalicki and Marcin Białasik. Categories of relational structures. In F. Parisi-Presicce, editor, *WADT'97, 12th workshop on Recent Trends in Algebraic Development Techniques*, volume 1376 of *LNCS*. Springer, 1998.

[WG96] Annika Wagner and Martin Gogolla. Defining operational behaviour of object specifications by attributed graph transformations. *Fundamenta Informaticae*, 26:407–431, 1996.

[WG99] Annika Wagner and Martin Gogolla. Semantics of object-oriented languages. In Ehrig et al. [EEKR99], chapter 4, pages 181–211.

[Wol98] Frank Wolter. On logics with coimplication. *Journal of Philosophical Logic*, 27(4):353–387, 1998.

[Zie88] Hans Zierer. *Programmierung mit Funktionsobjekten: Konstruktive Erzeugung semantischer Bereiche und Anwendung auf die partielle Auswertung.* Dissertation, Technische Univ. München, Fakultät für Informatik, 1988. Report TUM-I8803.

[Zie91] Hans Zierer. Relation-algebraic domain constructions. *Theoretical Computer Science*, 87:163–188, 1991.

## Appendix A

# Proofs of Auxiliary Properties

## A.1 Allegory Properties

**Proposition A.1.1** [←67] Both *modal rules*

$$\begin{array}{rcll} Q;R \sqcap S & \sqsubseteq & Q;(R \sqcap Q^{\smile};S) & \quad (\mathrm{m1}) \\ Q;R \sqcap S & \sqsubseteq & (Q \sqcap S;R^{\smile});R & \quad (\mathrm{m2}) \end{array}$$

together are equivalent to the *Dedekind rule*

$$Q;R \sqcap S \sqsubseteq (Q \sqcap S;R^{\smile});(R \sqcap Q^{\smile};S) \ .$$

**Proof**: The modal rules follow immediately from the Dedekind rule:

$$Q;R \sqcap S \sqsubseteq (Q \sqcap S;R^{\smile});(R \sqcap Q^{\smile};S) \sqsubseteq \begin{cases} (Q \sqcap S;R^{\smile});R \\ Q;(R \sqcap Q^{\smile};S) \end{cases}$$

Conversely, assume that the modal rules hold. Then we have

$$\begin{array}{rcll} Q;R \sqcap S & \sqsubseteq & Q;(R \sqcap Q^{\smile};S) \sqcap S & \quad (\mathrm{m1}) \\ & \sqsubseteq & (Q \sqcap S;(R \sqcap Q^{\smile};S)^{\smile});(R \sqcap Q^{\smile};S) & \quad (\mathrm{m2}) \\ & \sqsubseteq & (Q \sqcap S;R^{\smile});(R \sqcap Q^{\smile};S) \ . & \quad \text{meet properties: } R \sqcap Q^{\smile};S \sqsubseteq R \end{array}$$ □

**Lemma A.1.2** [←74, 76, 88, 91, 93, 100, 102, 129, 171, 176, 190–192] In every allegory, the following hold:

(i) If $F$ is univalent, then $F;(R \sqcap S) = F;R \sqcap F;S$

(ii) If $F$ is univalent, then $(R;F^{\smile} \sqcap S);F = R \sqcap S;F$

(iii) If $F$ is a mapping (i.e., total and univalent), then:

$$\begin{array}{rclcrcl} R;F & \sqsubseteq & S & \Leftrightarrow & R & \sqsubseteq & S;F^{\smile} \\ F^{\smile};R & \sqsubseteq & S & \Leftrightarrow & R & \sqsubseteq & F;S \end{array}$$

**Proof**:

(i) "$\sqsubseteq$" follows from meet-subdistributivity. For "$\sqsubseteq$" we use a modal rule and univalence of $F$:

$$F;R \sqcap F;S \sqsubseteq F;(R \sqcap F^{\smile};F;S) \sqsubseteq F;(R \sqcap S)$$

(ii) "$\sqsupseteq$" is an instance of a modal rule. For "$\sqsubseteq$" we use univalence of $F$:

$$(R;F^{\smile} \sqcap S);F \sqsubseteq R;F^{\smile};F \sqcap S;F \sqsubseteq R \sqcap S;F$$

(iii) "⇒": since $F$ is total, we have: $R \sqsubseteq R;F;F^{\smile} \sqsubseteq S;F^{\smile}$

"⇐": since $F$ is univalent, we have: $R;F \sqsubseteq S;F^{\smile};F \sqsubseteq S$

The second statement follows by conversion. □

**Lemma A.1.3** If $\mathbb{T}_{\mathcal{A},\mathcal{A}}$ and $\mathbb{T}_{\mathcal{A},\mathcal{B}}$ exist, we always have: $\mathbb{T}_{\mathcal{A},\mathcal{A}};\mathbb{T}_{\mathcal{A},\mathcal{B}} = \mathbb{T}_{\mathcal{A},\mathcal{B}}$.

**Proof**: "$\sqsubseteq$" is immediate since $\mathbb{T}_{\mathcal{A},\mathcal{B}}$ is the greatest element of $\mathsf{Mor}[\mathcal{A},\mathcal{B}]$.
For "$\sqsupseteq$", we know $\mathbb{I}_{\mathcal{A}} \sqsubseteq \mathbb{T}_{\mathcal{A},\mathcal{A}}$, so we obtain:

$$\mathbb{T}_{\mathcal{A},\mathcal{B}} = \mathbb{I}_{\mathcal{A}};\mathbb{T}_{\mathcal{A},\mathcal{B}} \sqsubseteq \mathbb{T}_{\mathcal{A},\mathcal{A}};\mathbb{T}_{\mathcal{A},\mathcal{B}}$$ □

In the presence of universal relations, totality may equivalently be defined in an alternative way which is frequently easier to handle in proofs:

**Lemma A.1.4** [←68] If for two objects $\mathcal{A}$ and $\mathcal{B}$ of an allegory, the universal relations $\mathbb{T}_{\mathcal{A},\mathcal{A}}$ and $\mathbb{T}_{\mathcal{A},\mathcal{B}}$ exist, then for every relation $R : \mathcal{A} \leftrightarrow \mathcal{B}$, the following three conditions are equivalent:

(i) $\mathbb{I}_{\mathcal{A}} \sqsubseteq R;R^{\smile}$
(ii) $\mathbb{T}_{\mathcal{A},\mathcal{A}} \sqsubseteq R;\mathbb{T}_{\mathcal{B},\mathcal{A}}$
(iii) $\mathbb{T}_{\mathcal{A},\mathcal{C}} \sqsubseteq R;\mathbb{T}_{\mathcal{B},\mathcal{C}}$ for all objects $\mathcal{C}$ for which $\mathbb{T}_{\mathcal{A},\mathcal{C}}$ and $\mathbb{T}_{\mathcal{B},\mathcal{C}}$ exist.

**Proof**: iii) ⇒ ii) is trivial. ii) ⇒ i) follows with a modal rule:

$$\mathbb{I}_{\mathcal{A}} = \mathbb{I}_{\mathcal{A}} \sqcap \mathbb{T}_{\mathcal{A},\mathcal{A}} \sqsubseteq \mathbb{I}_{\mathcal{A}} \sqcap R;\mathbb{T}_{\mathcal{B},\mathcal{A}} \sqsubseteq R;(R^{\smile};\mathbb{I}_{\mathcal{A}} \sqcap \mathbb{T}_{\mathcal{B},\mathcal{A}}) = R;R^{\smile}$$

i) ⇒ iii): For any object $\mathcal{C}$, we have: $\mathbb{T}_{\mathcal{A},\mathcal{C}} = \mathbb{I}_{\mathcal{A}};\mathbb{T}_{\mathcal{A},\mathcal{C}} \sqsubseteq R;R^{\smile};\mathbb{T}_{\mathcal{A},\mathcal{C}} \sqsubseteq R;\mathbb{T}_{\mathcal{B},\mathcal{C}}$ . □

We do not mention use of this lemma when proving or using totality in this way.

**Lemma A.1.5** [←69, 76, 92, 168] For three objects $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ in a locally co-complete allegory, let $\mathcal{R}$ be a set of relations from $\mathcal{B}$ to $\mathcal{C}$, and let $F : \mathcal{A} \leftrightarrow \mathcal{B}$ be a univalent relation. If $\mathcal{R}$ is non-empty or if $F$ is total, then

$$\bigsqcap\{R : \mathcal{R} \bullet F;R\} \sqsubseteq F;\bigsqcap\mathcal{R}$$

(Equality then follows from co-completeness.)

**Proof**: First assume that $\mathcal{R}$ is non-empty and $R_0 \in \mathcal{R}$. Then:

$$\begin{array}{rcll} \bigsqcap\{R : \mathcal{R} \bullet F;R\} & = & F;R_0 \sqcap \bigsqcap\{R : \mathcal{R} \bullet F;R\} & \\ & \sqsubseteq & F;(R_0 \sqcap F^{\smile};\bigsqcap\{R : \mathcal{R} \bullet F;R\}) & \text{modal rule} \\ & \sqsubseteq & F;(R_0 \sqcap \bigsqcap\{R : \mathcal{R} \bullet F^{\smile};F;R\}) & \text{co-completeness} \\ & \sqsubseteq & F;(R_0 \sqcap \bigsqcap\{R : \mathcal{R} \bullet R\}) & F \text{ univalent} \\ & = & F;\bigsqcap\mathcal{R} & \end{array}$$

If $\mathcal{R}$ is empty, then, using totality of $F$:

$$\bigsqcap\{R : \mathcal{R} \bullet F;R\} = \bigsqcap\emptyset = \mathbb{T} = F;\mathbb{T} = F;\bigsqcap\emptyset = F;\bigsqcap\mathcal{R}$$ □



## A.2 Partial Identities

With respect to partial identities, in all allegories, the following laws hold (some of them can be used for an alternative axiomatisation of allegories, see [Gut99, DG00]):

**Lemma A.2.1** [←168]

(i) $(\mathbb{I} \sqcap R);(\mathbb{I} \sqcap S) = \mathbb{I} \sqcap R \sqcap S$
(ii) $(\mathbb{I} \sqcap R);(\mathbb{I} \sqcap S) = (\mathbb{I} \sqcap S);(\mathbb{I} \sqcap R)$
(iii) $Q \sqcap R;(\mathbb{I} \sqcap S) = (Q \sqcap R);(\mathbb{I} \sqcap S)$
(iv) $\mathbb{I} \sqcap Q;(R \sqcap S) = \mathbb{I} \sqcap (Q \sqcap R^{\smile});S$
(v) $Q \sqcap Q;S;S^{\smile} = Q;(\mathbb{I} \sqcap S;S^{\smile})$

**Proof**: (i)
$$\begin{array}{rcll} (\mathbb{I} \sqcap R);(\mathbb{I} \sqcap S) & \sqsubseteq & \mathbb{I};(\mathbb{I} \sqcap S) \sqcap R;(\mathbb{I} \sqcap S) & \\ & \sqsubseteq & \mathbb{I} \sqcap R \sqcap S & \\ & = & (\mathbb{I} \sqcap R);\mathbb{I} \sqcap S & \\ & \sqsubseteq & (\mathbb{I} \sqcap R);(\mathbb{I} \sqcap (\mathbb{I} \sqcap R)^{\smile};S) & \text{modal rule} \\ & \sqsubseteq & (\mathbb{I} \sqcap R);(\mathbb{I} \sqcap S) & \end{array}$$

(ii) follows from (i) together with commutativity of meet.

(iii)
$$\begin{array}{rcll} Q \sqcap R;(\mathbb{I} \sqcap S) & \sqsubseteq & (Q;(\mathbb{I} \sqcap S)^{\smile} \sqcap R);(\mathbb{I} \sqcap S) & \text{modal rule} \\ & \sqsubseteq & (Q \sqcap R);(\mathbb{I} \sqcap S) & \\ & \sqsubseteq & Q;(\mathbb{I} \sqcap S) \sqcap R;(\mathbb{I} \sqcap S) \sqsubseteq Q \sqcap R;(\mathbb{I} \sqcap S) & \end{array}$$

(iv)
$$\begin{array}{rcll} \mathbb{I} \sqcap Q;(R \sqcap S) & = & \mathbb{I} \sqcap (Q \sqcap \mathbb{I};(R \sqcap S)^{\smile});(R \sqcap S) & \text{modal rule} \\ & = & \mathbb{I} \sqcap (Q \sqcap R^{\smile} \sqcap S^{\smile});(R \sqcap S) & \\ & \sqsubseteq & \mathbb{I} \sqcap (Q \sqcap R^{\smile});S & \end{array}$$

The opposite inclusion is shown analogously.

(v)
$$\begin{array}{rcll} Q \sqcap Q;S;S^{\smile} & = & Q;\mathbb{I} \sqcap Q;S;S^{\smile} & \\ & \sqsubseteq & Q;(\mathbb{I} \sqcap Q^{\smile};Q;S;S^{\smile}) & \text{modal rule} \\ & = & Q;(\mathbb{I} \sqcap (S \sqcap Q^{\smile};Q;S);S^{\smile}) & \text{(iv)} \\ & \sqsubseteq & Q;(\mathbb{I} \sqcap S;S^{\smile}) & \end{array}$$

The opposite inclusion is trivial. □

**Lemma A.2.2** [←102, 108, 169, 187]

(i) $\mathsf{dom}\,((Q \sqcap R);S) = \mathbb{I} \sqcap Q;\mathsf{dom}\,S;R^{\smile}$ and $\mathsf{ran}\,(Q;(R \sqcap S)) = \mathbb{I} \sqcap R^{\smile};\mathsf{ran}\,Q;S$
(ii) $\mathsf{dom}\,(Q;S) = \mathsf{dom}\,(Q;\mathsf{dom}\,S)$ and $\mathsf{ran}\,(Q;S) = \mathsf{ran}\,(\mathsf{ran}\,Q;S)$
(iii) $\mathsf{dom}\,(\mathsf{dom}\,R;S) = \mathsf{dom}\,R \sqcap \mathsf{dom}\,S$ and $\mathsf{ran}\,(R;\mathsf{ran}\,S) = \mathsf{ran}\,R \sqcap \mathsf{ran}\,S$
(iv) $\mathsf{dom}\,(Q \sqcap R;S) = \mathsf{dom}\,(Q;S^{\smile} \sqcap R)$
(v) $\mathsf{dom}\,(Q \sqcap R) = \mathbb{I} \sqcap Q;R^{\smile}$

**Proof**: We only show the proofs for the respective first equalities:

(i)
$$\begin{array}{rcll} \mathsf{dom}\,((Q \sqcap R);S) & = & \mathbb{I} \sqcap (Q \sqcap R);S;S^\smile;(Q^\smile \sqcap R^\smile) & \\ & = & \mathbb{I} \sqcap ((Q \sqcap R);S;S^\smile \sqcap Q));R^\smile & \text{Lemma A.2.1.iv)} \\ & = & \mathbb{I} \sqcap ((Q \sqcap R);S;S^\smile \sqcap (Q \sqcap R));R^\smile & \text{Lemma A.2.1.iv)} \\ & = & \mathbb{I} \sqcap (Q \sqcap R);(\mathbb{I} \sqcap S;S^\smile);R^\smile & \text{Lemma A.2.1.v)} \\ & = & \mathbb{I} \sqcap (Q;(\mathbb{I} \sqcap S;S^\smile) \sqcap R);R^\smile & \text{Lemma A.2.1.iii)} \\ & = & \mathbb{I} \sqcap Q;(\mathbb{I} \sqcap S;S^\smile);R^\smile & \text{Lemma A.2.1.iv)} \\ & = & \mathbb{I} \sqcap Q;\mathsf{dom}\,S;R^\smile & \text{Def. } \mathsf{dom} \end{array}$$

(ii)
$$\begin{array}{rcll} \mathsf{dom}\,(Q;S) & = & \mathsf{dom}\,((Q \sqcap Q);S) & \\ & = & \mathbb{I} \sqcap Q;\mathsf{dom}\,S;Q^\smile & \text{(i)} \\ & = & \mathbb{I} \sqcap Q;\mathsf{dom}\,S;\mathsf{dom}\,S^\smile;Q^\smile & \\ & = & \mathsf{dom}\,(Q;\mathsf{dom}\,S) & \text{Def. } \mathsf{dom} \end{array}$$

(iii) With (ii): $\mathsf{dom}\,(\mathsf{dom}\,R;S) = \mathsf{dom}\,(\mathsf{dom}\,R;\mathsf{dom}\,S) = \mathsf{dom}\,R;\mathsf{dom}\,S = \mathsf{dom}\,R \sqcap \mathsf{dom}\,S$

(iv)
$$\begin{array}{rcll} \mathsf{dom}\,(Q \sqcap R;S) & \sqsubseteq & \mathsf{dom}\,((Q;S^\smile \sqcap R);S) & \text{modal rule} \\ & = & \mathsf{dom}\,((Q;S^\smile \sqcap R);\mathsf{dom}\,S) & \text{(ii)} \\ & \sqsubseteq & \mathsf{dom}\,(Q;S^\smile \sqcap R) & \end{array}$$

The opposite inclusion is shown analogously.

(v) With (i): $\mathsf{dom}\,(Q \sqcap R) = \mathsf{dom}\,((Q \sqcap R);\mathbb{I}) = \mathbb{I} \sqcap Q;\mathsf{dom}\,\mathbb{I};R^\smile = \mathbb{I} \sqcap Q;R^\smile$ □

**Lemma A.2.3** [←76] $(P \sqcap Q) \times (R \sqcap S) = (P \times R) \sqcap (Q \times S)$, and analogously, given two $\mathcal{I}$-indexed families $(R_i)_{i:\mathcal{I}}$ and $(S_i)_{i:\mathcal{I}}$ with $R_i : \mathcal{A} \leftrightarrow \mathcal{C}$ and $S_i : \mathcal{B} \leftrightarrow \mathcal{D}$ for all $i : \mathcal{I}$,

$$\bigsqcap\{i : \mathcal{I} \bullet R_i\} \times \bigsqcap\{i : \mathcal{I} \bullet S_i\} = \bigsqcap\{i : \mathcal{I} \bullet R_i \times S_i\}$$

**Proof**:

$$\begin{array}{rcll} (P \sqcap Q) \times (R \sqcap S) & = & \pi;(P \sqcap Q);\pi^\smile \sqcap \rho;(R \sqcap S);\rho^\smile & \\ & = & \pi;P;\pi^\smile \sqcap \pi;Q;\pi^\smile \sqcap \rho;R;\rho^\smile \sqcap \rho;S;\rho^\smile & \pi,\ \rho \text{ univalent} \\ & = & (P \times R) \sqcap (Q \times S) & \end{array}$$

$$\begin{array}{cll} & \bigsqcap\{i : \mathcal{I} \bullet R_i\} \times \bigsqcap\{i : \mathcal{I} \bullet S_i\} & \\ = & \pi;\bigsqcap\{i : \mathcal{I} \bullet R_i\};\pi^\smile \sqcap \rho;\bigsqcap\{i : \mathcal{I} \bullet S_i\};\rho^\smile & \\ = & \bigsqcap\{i : \mathcal{I} \bullet \pi;R_i;\pi^\smile\} \sqcap \bigsqcap\{i : \mathcal{I} \bullet \rho;S_i;\rho^\smile\} & \text{Lemma A.1.5} \\ = & \bigsqcap(\{i : \mathcal{I} \bullet \pi;R_i;\pi^\smile\} \cup \{i : \mathcal{I} \bullet \rho;S_i;\rho^\smile\}) & \text{meet comm. and assoc.} \\ = & \bigsqcap\{i : \mathcal{I} \bullet \pi;R_i;\pi^\smile \sqcap \rho;S_i;\rho^\smile\} & \text{meet comm. and assoc.} \\ = & \bigsqcap\{i : \mathcal{I} \bullet R_i \times S_i\} & \end{array}$$
□



Note that this property crucially relies on the associativity of the meets in the product factors with the meet in the definition of the relational product $\times$. Therefore, the analogous property for joins cannot be shown; in general, we have

$$(P \sqcup Q) \times (R \sqcup S) \not\sqsubseteq (P \times R) \sqcup (Q \times S) \ .$$

What can be shown is the opposite inclusion, or even the equality

$$(P \sqcup Q) \times (R \sqcup S) = (P \times R) \sqcup (P \times S) \sqcup (Q \times R) \sqcup (Q \times S) \ ;$$

however, this is not very useful in the contexts where we might seem to need such a property.

## A.3 Dedekind Category Properties

**Lemma A.3.1** [←86] For the domain of the univalent part of $R : \mathcal{A} \leftrightarrow \mathcal{B}$, we have: $\mathsf{dom}\,(\mathsf{upa}\,R) = \mathsf{dom}\,R \sqcap \mathsf{dom}\,(R^\smile \backslash \mathbb{I})$.

Furthermore, for all $Y \sqsubseteq \mathsf{dom}\,R$ we have

$$Y;R \text{ is univalent} \qquad \Leftrightarrow \qquad Y \sqsubseteq \mathsf{dom}\,(\mathsf{upa}\,R)$$

**Proof**: The inclusion $\mathsf{dom}\,(\mathsf{upa}\,R) = \mathsf{dom}\,(R \sqcap R^\smile \backslash \mathbb{I}) \sqsubseteq \mathsf{dom}\,R \sqcap \mathsf{dom}\,(R^\smile \backslash \mathbb{I})$ is obvious. The opposite inclusion is obtained in the following way:

$$\begin{array}{rcll} \mathsf{dom}\,R \sqcap \mathsf{dom}\,(R^\smile \backslash \mathbb{I}) & = & (\mathbb{I} \sqcap R;R^\smile) \sqcap (\mathbb{I} \sqcap (R^\smile \backslash \mathbb{I});(R^\smile \backslash \mathbb{I})^\smile) & \\ & = & \mathbb{I} \sqcap R;R^\smile \sqcap (R^\smile \backslash \mathbb{I});(\mathbb{I}/R) & \\ & \sqsubseteq & \mathbb{I} \sqcap R;(R^\smile \sqcap R^\smile;(R^\smile \backslash \mathbb{I});(\mathbb{I}/R)) & \text{modal rule} \\ & \sqsubseteq & \mathbb{I} \sqcap R;R^\smile;(R^\smile \backslash \mathbb{I});(\mathbb{I}/R) & \\ & \sqsubseteq & \mathbb{I} \sqcap R;\mathbb{I};(\mathbb{I}/R) & \text{residual} \\ & = & \mathbb{I} \sqcap R;(\mathbb{I}/R) & \\ & = & \mathbb{I} \sqcap R;(R^\smile \backslash \mathbb{I})^\smile & \\ & = & \mathsf{dom}\,(R \sqcap R^\smile \backslash \mathbb{I}) & \text{Lemma A.2.2.v)} \\ & = & \mathsf{dom}\,(\mathsf{upa}\,R) & \end{array}$$

Furthermore, we have:

$$\begin{array}{rcl} (\mathsf{dom}\,R);(R^\smile \backslash \mathbb{I}) & \sqsubseteq & R;R^\smile;(R^\smile \backslash \mathbb{I}) \sqsubseteq R;\mathbb{I} = R \\ (\mathsf{dom}\,(R^\smile \backslash \mathbb{I}));R & \sqsubseteq & (R^\smile \backslash \mathbb{I});(R^\smile \backslash \mathbb{I})^\smile;R = (R^\smile \backslash \mathbb{I});(\mathbb{I}/R);R \sqsubseteq (R^\smile \backslash \mathbb{I});\mathbb{I} = (R^\smile \backslash \mathbb{I}) \end{array}$$

Then, assuming $Y \sqsubseteq \mathsf{dom}\,R$, we have:

$$
\begin{array}{lll}
& Y;R \text{ is univalent} & \\
\Leftrightarrow & R^{\smile};Y;R \sqsubseteq \mathbb{I} & \\
\Leftrightarrow & Y;R \sqsubseteq R^{\smile}\backslash\mathbb{I} & \\
\Leftrightarrow & Y \sqsubseteq \mathsf{dom}\,(R^{\smile}\backslash\mathbb{I}) & \text{"}\Leftarrow\text{": } \mathsf{dom}\,(R^{\smile}\backslash\mathbb{I});R \sqsubseteq (R^{\smile}\backslash\mathbb{I}) \text{ from above} \\
& & \text{"}\Rightarrow\text{": } Y \sqsubseteq \mathsf{dom}\,R \\
\Leftrightarrow & Y \sqsubseteq \mathsf{dom}\,R \sqcap \mathsf{dom}\,(R^{\smile}\backslash\mathbb{I}) & Y \sqsubseteq \mathsf{dom}\,R \\
\Leftrightarrow & Y \sqsubseteq \mathsf{dom}\,(R \sqcap R^{\smile}\backslash\mathbb{I}) & \\
\Leftrightarrow & Y \sqsubseteq \mathsf{dom}\,(\mathsf{upa}\,R) &
\end{array}
$$

□

## A.4 Semi-Complements and Partial Identities in Dedekind Categories

**Lemma A.4.1** [←171]

(i) If $F$ is univalent, then

$$(\mathsf{ran}\,R)^{\sim} = \mathsf{ran}\,(((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,(R;F^{\smile}));F) \sqcup (\mathsf{ran}\,R \sqcup \mathsf{ran}\,F)^{\sim} \quad ,$$

(ii) If $F$ is total and univalent, then $(\mathsf{ran}\,(R;F^{\smile}))^{\sim} \sqsubseteq \mathsf{ran}\,((\mathsf{ran}\,R)^{\sim};F^{\smile})$,

**Proof**:

(i)
$$
\begin{array}{lll}
(\mathsf{ran}\,R)^{\sim} \sqsubseteq Y & \Leftrightarrow & \mathbb{I} \sqsubseteq Y \sqcup \mathsf{ran}\,R \\
& \Leftrightarrow & \mathsf{ran}\,F \sqcup (\mathsf{ran}\,F)^{\sim} \sqsubseteq Y \sqcup \mathsf{ran}\,R \\
& \Leftrightarrow & \mathsf{ran}\,F \sqsubseteq Y \sqcup \mathsf{ran}\,R \quad \text{and} \quad (\mathsf{ran}\,F)^{\sim} \sqsubseteq Y \sqcup \mathsf{ran}\,R
\end{array}
$$

For the second inclusion, we have:

$$
\begin{array}{lll}
(\mathsf{ran}\,F)^{\sim} \sqsubseteq Y \sqcup \mathsf{ran}\,R & \Leftrightarrow & \mathbb{I} \sqsubseteq Y \sqcup \mathsf{ran}\,R \sqcup \mathsf{ran}\,F \\
& \Leftrightarrow & (\mathsf{ran}\,R \sqcup \mathsf{ran}\,F)^{\sim} \sqsubseteq Y
\end{array}
$$

For the first inclusion:

$$
\begin{array}{llll}
\mathsf{ran}\,F \sqsubseteq Y \sqcup \mathsf{ran}\,R & \Leftrightarrow & F^{\smile};F \sqsubseteq Y \sqcup \mathsf{ran}\,R & F \text{ univalent} \\
& \Leftrightarrow & \mathsf{dom}\,F \sqsubseteq F;Y;F^{\smile} \sqcup F;\mathsf{ran}\,R;F^{\smile} & F \text{ univalent} \\
& \Leftrightarrow & \mathsf{dom}\,F \sqsubseteq F;Y;F^{\smile} \sqcup \mathsf{ran}\,(R;F^{\smile}) & F \text{ univalent} \\
& \Leftrightarrow & (\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,(R;F^{\smile}) \sqsubseteq F;Y;F^{\smile} & \\
& \Leftrightarrow & F^{\smile};((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,(R;F^{\smile}));F \sqsubseteq Y & F \text{ univalent} \\
& \Leftrightarrow & \mathsf{ran}\,(((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,(R;F^{\smile}));F) \sqsubseteq Y & F \text{ univalent}
\end{array}
$$

(ii) First we have:

$$
\begin{array}{llll}
(\mathsf{ran}\,R)^{\sim} & = & \mathsf{ran}\,(((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,(R;F^{\smile}));F) \sqcup (\mathsf{ran}\,R \sqcup \mathsf{ran}\,F)^{\sim} & \\
& = & \mathsf{ran}\,((\mathsf{ran}\,(R;F^{\smile}))^{\sim};F) \sqcup (\mathsf{ran}\,R \sqcup \mathsf{ran}\,F)^{\sim} & \mathsf{dom}\,F = \mathbb{I} \\
& \sqsupseteq & \mathsf{ran}\,((\mathsf{ran}\,(R;F^{\smile}))^{\sim};F) & \\
& = & F^{\smile};(\mathsf{ran}\,(R;F^{\smile}))^{\sim};F & F \text{ univalent}
\end{array}
$$

The whole inclusion is then equivalent to: $(\mathsf{ran}\,(R;F^{\smile}))^{\sim} \sqsubseteq F;(\mathsf{ran}\,R)^{\sim};F^{\smile}$. □



**Lemma A.4.2** [←179]

(i) If $F$ is total, univalent and surjective, then

$$(\mathsf{ran}\,R)^{\sim} = \mathsf{ran}\,((\mathsf{ran}\,(R;F^{\smile}))^{\sim};F) \qquad \text{and} \qquad (\mathsf{ran}\,(R;F^{\smile}))^{\sim} \sqsubseteq \mathsf{ran}\,((\mathsf{ran}\,R)^{\sim};F^{\smile})$$

(ii) If $F$ is univalent and injective, then

$$(\mathsf{ran}\,(R;F))^{\sim} = \mathsf{ran}\,(((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,R);F) \sqcup (\mathsf{ran}\,F)^{\sim}$$

(iii) If $F$ is total, univalent and injective, then

$$(\mathsf{ran}\,(R;F))^{\sim} = \mathsf{ran}\,((\mathsf{ran}\,R)^{\sim};F) \sqcup (\mathsf{ran}\,F)^{\sim}$$

**Proof**:

(i) We start from Lemma A.4.1.i):

$$
\begin{array}{lll}
& (\mathsf{ran}\,R)^{\sim} & \\
= & \mathsf{ran}\,(((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,(R;F^{\smile}));F) \sqcup (\mathsf{ran}\,R \sqcup \mathsf{ran}\,F)^{\sim} & \\
= & \mathsf{ran}\,((\mathsf{ran}\,(R;F^{\smile}))^{\sim};F) & F \text{ surj.: } (\mathsf{ran}\,F)^{\sim} = \top\!\!\top \\
= & F^{\smile};(\mathsf{ran}\,(R;F^{\smile}))^{\sim};F & F \text{ univalent}
\end{array}
$$

With Lemma A.1.2.iii) we then obtain: $(\mathsf{ran}\,(R;F^{\smile}))^{\sim} \sqsubseteq F;(\mathsf{ran}\,R)^{\sim};F^{\smile}$.

(ii)
$$
\begin{array}{lll}
(\mathsf{ran}\,(R;F))^{\sim} \sqsubseteq Y & \Leftrightarrow & \mathbb{I} \sqsubseteq Y \sqcup \mathsf{ran}\,(R;F) \\
& \Leftrightarrow & \mathsf{ran}\,F \sqcup (\mathsf{ran}\,F)^{\sim} \sqsubseteq Y \sqcup \mathsf{ran}\,(R;F) \\
& \Leftrightarrow & \mathsf{ran}\,F \sqsubseteq Y \sqcup \mathsf{ran}\,(R;F) \quad \wedge \quad (\mathsf{ran}\,F)^{\sim} \sqsubseteq Y \sqcup \mathsf{ran}\,(R;F)
\end{array}
$$

The second inclusion easily resolves in the following way:

$$
\begin{array}{lll}
& (\mathsf{ran}\,F)^{\sim} \sqsubseteq Y \sqcup \mathsf{ran}\,(R;F) & \\
\Leftrightarrow & \mathbb{I} \sqsubseteq Y \sqcup \mathsf{ran}\,(\mathsf{ran}\,R;F) \sqcup \mathsf{ran}\,F & \\
\Leftrightarrow & \mathbb{I} \sqsubseteq Y \sqcup \mathsf{ran}\,F & \mathsf{ran}\,(\mathsf{ran}\,R;F) \sqsubseteq \mathsf{ran}\,F \\
\Leftrightarrow & (\mathsf{ran}\,F)^{\sim} \sqsubseteq Y &
\end{array}
$$

For the first inclusion, we calculate:

$$
\begin{array}{lll}
& \mathsf{ran}\,F \sqsubseteq Y \sqcup \mathsf{ran}\,(R;F) & \\
\Leftrightarrow & \mathsf{ran}\,F \sqsubseteq \mathsf{ran}\,F;Y;\mathsf{ran}\,F \sqcup \mathsf{ran}\,(R;F) & \\
\Leftrightarrow & \mathsf{ran}\,F \sqsubseteq F^{\smile};F;Y;F^{\smile};F \sqcup F^{\smile};\mathsf{ran}\,R;F & \\
\Leftrightarrow & F^{\smile};\mathsf{dom}\,F;F \sqsubseteq F^{\smile};(F;Y;F^{\smile} \sqcup \mathsf{ran}\,R);F & \\
\Leftrightarrow & \mathsf{dom}\,F \sqsubseteq F;Y;F^{\smile} \sqcup \mathsf{ran}\,R & F \text{ injective} \\
\Leftrightarrow & (\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,R \sqsubseteq F;Y;F^{\smile} & \\
\Leftrightarrow & F^{\smile};((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,R);F \sqsubseteq Y & F \text{ univalent and injective} \\
\Leftrightarrow & \mathsf{ran}\,(((\mathsf{dom}\,F) \smallsetminus \mathsf{ran}\,R);F) \sqsubseteq Y & F \text{ univalent}
\end{array}
$$

(iii) follows immediately from (ii). □

## A.5 Symmetric Quotients

The following properties of symmetric quotients are shown here using only the setting of locally complete distributive allegories.

**Lemma A.5.1** [←125]

(i) $(\mathrm{syq}(R,S))^{\smile} = \mathrm{syq}(S,R)$

(ii) $R;\mathrm{syq}(R,S) = S;\mathsf{ran}\,(\mathrm{syq}(R,S))$

(iii) $\mathrm{syq}(R,S);\mathsf{ran}\,S \sqsubseteq R^{\smile};S$

**Proof**:

(i)
$$\begin{aligned}(\mathrm{syq}(R,S))^{\smile} &= \bigsqcup\{X \mid Q;X \sqsubseteq S \text{ and } X;S^{\smile} \sqsubseteq Q^{\smile} \bullet X^{\smile}\}\\ &= \bigsqcup\{Y \mid Q;Y^{\smile} \sqsubseteq S \text{ and } Y^{\smile};S^{\smile} \sqsubseteq Q^{\smile}\}\\ &= \bigsqcup\{Y \mid S;Y \sqsubseteq Q \text{ and } Y;Q^{\smile} \sqsubseteq S^{\smile}\} = \mathrm{syq}(S,R)\end{aligned}$$

(ii)
$$\begin{aligned}R;\mathrm{syq}(R,S) &= R;\mathrm{syq}(R,S);\mathsf{ran}\,(\mathrm{syq}(R,S))\\ &\sqsubseteq S;\mathsf{ran}\,(\mathrm{syq}(R,S)) && \text{Def. } \mathsf{syq}\\ &\sqsubseteq S;(\mathrm{syq}(R,S))^{\smile};\mathrm{syq}(R,S) && \text{Def. } \mathsf{ran}\\ &= S;\mathrm{syq}(S,R);\mathrm{syq}(R,S) && \text{(i)}\\ &\sqsubseteq R;\mathrm{syq}(R,S) && \text{Def. } \mathsf{syq}\end{aligned}$$

(iii) $\mathrm{syq}(R,S);\mathsf{ran}\,S \sqsubseteq \mathrm{syq}(R,S);S^{\smile};S \sqsubseteq R^{\smile};S$ □

**Lemma A.5.2** [←125] If $R$ is difunctional, then $R^{\smile};R;\mathrm{syq}(R,S) \sqsubseteq \mathrm{syq}(R,S)$, and $R^{\smile};R;\mathrm{syq}(R,S) = \mathsf{ran}\,R;\mathrm{syq}(R,S)$.

**Proof**:
$$\begin{aligned}X \sqsubseteq \mathrm{syq}(R,S) \quad&\Leftrightarrow\quad R;X \sqsubseteq S \text{ and } X;S^{\smile} \sqsubseteq R^{\smile}\\ &\Rightarrow\quad R;X \sqsubseteq S \text{ and } R^{\smile};R;X;S^{\smile} \sqsubseteq R^{\smile};R;R^{\smile}\\ &\Leftrightarrow\quad R;R^{\smile};R;X \sqsubseteq S \text{ and } R^{\smile};R;X;S^{\smile} \sqsubseteq R^{\smile}\\ &\Leftrightarrow\quad R^{\smile};R;X \sqsubseteq \mathrm{syq}(R,S)\end{aligned}$$

The second statement easily follows from this:

$$R^{\smile};R;\mathrm{syq}(R,S) = \mathsf{ran}\,R;R^{\smile};R;\mathrm{syq}(R,S) \sqsubseteq \mathsf{ran}\,R;\mathrm{syq}(R,S) \sqsubseteq R^{\smile};R;\mathrm{syq}(R,S)$$ □

# Appendix B

# Proofs for Relational Rewriting, Chapter 6

## B.1 Correctness of the Glued Tabulation Construction

For showing Theorem 6.2.6, let us first summarise the properties resulting from the sub-constructions of Def. 6.2.5:

- $C_1 \xrightarrow{\mathrm{X}_u} G_u \xleftarrow{\Psi_u} C_2$ is a gluing for $\lambda_1;U;\lambda_2^{\smile}$:

$$\begin{aligned}\mathrm{X}_u;\Psi_u^{\smile} &= (\lambda_1;U;\lambda_2^{\smile})^{\boxast} = \lambda_1;U^{\boxast};\lambda_2^{\smile} & \lambda_1^{\smile};\mathrm{X}_u;\Psi_u^{\smile};\lambda_2 &= c_1;U^{\boxast};c_2\\ \mathrm{X}_u;\mathrm{X}_u^{\smile} &= (\lambda_1;U;\lambda_2^{\smile})^{\vartriangleright} = \lambda_1;U^{\vartriangleright};\lambda_1^{\smile} & \lambda_1^{\smile};\mathrm{X}_u;\mathrm{X}_u^{\smile};\lambda_1 &= c_1;U^{\vartriangleright};c_1\\ \Psi_u;\Psi_u^{\smile} &= (\lambda_1;U;\lambda_2^{\smile})^{\vartriangleleft} = \lambda_2;U^{\vartriangleleft};\lambda_2^{\smile} & \lambda_2^{\smile};\Psi_u;\Psi_u^{\smile};\lambda_2 &= c_2;U^{\vartriangleleft};c_2\\ \mathrm{X}_u^{\smile};\mathrm{X}_u \sqcup \Psi_u^{\smile};\Psi_u &= \mathbb{I}\end{aligned}$$

  (When using the porperties to the right, we immediately drop the $c_1$ respectively $c_2$ where they are imlied by the context.)

- $G_1 \xleftarrow{\mathrm{X}_v} G_v \xrightarrow{\Psi_v} G_2$ is a tabulation for $V$:

  $\mathrm{X}_v^{\smile};\Psi_v = V$ $\quad\mathrm{X}_v^{\smile};\mathrm{X}_v = \mathsf{dom}\,V$ $\quad\Psi_v^{\smile};\Psi_v = \mathsf{ran}\,V$ $\quad\mathrm{X}_v;\mathrm{X}_v^{\smile} \sqcap \Psi_v;\Psi_v^{\smile} = \mathbb{I}$

- $G_v \xrightarrow{\Lambda_v} G_3 \xleftarrow{\Lambda_u} G_u$ is a gluing for $B$:

  $\Lambda_v;\Lambda_u^{\smile} = B^{\boxast}$ $\quad\Lambda_v;\Lambda_v^{\smile} = B^{\vartriangleright}$ $\quad\Lambda_u;\Lambda_u^{\smile} = B^{\vartriangleleft}$ $\quad\Lambda_v^{\smile};\Lambda_v \sqcup \Lambda_u^{\smile};\Lambda_u = \mathbb{I}$

$$\begin{tikzcd}
C_2 \arrow[dr, "\Psi_u"'] \arrow[rr, phantom, "\text{gluing}" below] & & C_1 \arrow[ll, "\lambda_1;U;\lambda_2^{\smile}"'] \arrow[dl, "\mathrm{X}_u"] & & v_2 \arrow[rr, phantom, "\text{tabulation}" below] & & v_1 \arrow[ll, "V"'] \\
& G_u \arrow[drr, "\Lambda_u"'] \arrow[rrrr, phantom, "\text{gluing}" below] & & & & G_v \arrow[ul, "\Psi_v"] \arrow[ur, "\mathrm{X}_v"'] \arrow[llll, "B"'] \arrow[dll, "\Lambda_v"] & \\
& & & G_3 & & &
\end{tikzcd}$$

We first of all simplify the different compositions with $B$, in order to be able to eliminate $B^{\boxast}$, $B^{\vartriangleright}$, and $B^{\vartriangleleft}$ from later considerations.

$$\begin{aligned}\mathrm{X}_v^{\smile};B &= \mathrm{X}_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqcup \mathrm{X}_v^{\smile};\Psi_v;\lambda_2^{\smile};\Psi_u = \mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u \sqcup V;\lambda_2^{\smile};\Psi_u\\ \Psi_v^{\smile};B &= \Psi_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqcup \Psi_v^{\smile};\Psi_v;\lambda_2^{\smile};\Psi_u = V^{\smile};\lambda_1^{\smile};\mathrm{X}_u \sqcup \mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u\\ \mathrm{X}_u;B^{\smile} &= \mathrm{X}_u;\mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile} \sqcup \mathrm{X}_u;\Psi_u^{\smile};\lambda_2;\Psi_v^{\smile} = \lambda_1;U^{\vartriangleright};\mathrm{X}_v^{\smile} \sqcup \lambda_1;U^{\boxast};\Psi_v^{\smile}\\ \Psi_u;B^{\smile} &= \Psi_u;\mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile} \sqcup \Psi_u;\Psi_u^{\smile};\lambda_2;\Psi_v^{\smile} = \lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \lambda_2;U^{\vartriangleleft};\Psi_v^{\smile}\end{aligned}$$

$$
\begin{aligned}
\mathrm{X}_v^{\smile};B;B^{\smile} &= \mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u;B^{\smile} \sqcup V;\lambda_2^{\smile};\Psi_u;B^{\smile}\\
&= \mathsf{dom}\,V;\lambda_1^{\smile};\lambda_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \mathsf{dom}\,V;\lambda_1^{\smile};\lambda_1;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad V;\lambda_2^{\smile};\lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup V;\lambda_2^{\smile};\lambda_2;U^{\lhd};\Psi_v^{\smile}\\
&= \mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \mathsf{dom}\,V;c_1;U^{\boxast};\Psi_v^{\smile} \sqcup V;c_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup V;c_2;U^{\lhd};\Psi_v^{\smile}\\
&= \mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile}\\
\Psi_v^{\smile};B;B^{\smile} &= V^{\smile};\lambda_1^{\smile};\mathrm{X}_u;B^{\smile} \sqcup \mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u;B^{\smile}\\
&= V^{\smile};\lambda_1^{\smile};\lambda_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup V^{\smile};\lambda_1^{\smile};\lambda_1;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \mathsf{ran}\,V;\lambda_2^{\smile};\lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \mathsf{ran}\,V;\lambda_2^{\smile};\lambda_2;U^{\lhd};\Psi_v^{\smile}\\
&= V^{\smile};c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup V^{\smile};c_1;U^{\boxast};\Psi_v^{\smile} \sqcup \mathsf{ran}\,V;c_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile}\\
&= \mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile}\\
\mathrm{X}_u;B^{\smile};B &= \lambda_1;U^{\rhd};\mathrm{X}_v^{\smile};B \sqcup \lambda_1;U^{\boxast};\Psi_v^{\smile};B\\
&= \lambda_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u \sqcup \lambda_1;U^{\rhd};V;\lambda_2^{\smile};\Psi_u \sqcup\\
&\quad \lambda_1;U^{\boxast};V^{\smile};\lambda_1^{\smile};\mathrm{X}_u \sqcup \lambda_1;U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u\\
&= \lambda_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u \sqcup \lambda_1;U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u\\
\Psi_u;B^{\smile};B &= \lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile};B \sqcup \lambda_2;U^{\lhd};\Psi_v^{\smile};B\\
&= \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u \sqcup \lambda_2;(U^{\smile})^{\boxast};V;\lambda_2^{\smile};\Psi_u \sqcup\\
&\quad \lambda_2;U^{\lhd};V^{\smile};\lambda_1^{\smile};\mathrm{X}_u \sqcup \lambda_2;U^{\lhd};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u\\
&= \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u \sqcup \lambda_2;U^{\lhd};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u
\end{aligned}
$$

For $\mathrm{X}_u$ and $\Psi_u$, the third steps are contained in the first:

$$
\begin{aligned}
\mathrm{X}_u;B^{\smile};B;B^{\smile} &= \lambda_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u;B^{\smile} \sqcup \lambda_1;U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u;B^{\smile}\\
&= \lambda_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\lambda_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \lambda_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\lambda_1;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \lambda_1;U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \lambda_1;U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\lambda_2;U^{\lhd};\Psi_v^{\smile}\\
&= \lambda_1;U^{\rhd};\mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \lambda_1;U^{\rhd};\mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \lambda_1;U^{\boxast};\mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \lambda_1;U^{\boxast};\mathsf{ran}\,V;U^{\lhd};\Psi_v^{\smile}\\
&\sqsubseteq \lambda_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \lambda_1;U^{\boxast};\Psi_v^{\smile}\\
&= \mathrm{X}_u;B^{\smile}
\end{aligned}
$$

$$
\begin{aligned}
\Psi_u;B^{\smile};B;B^{\smile} &= \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u;B^{\smile} \sqcup \lambda_2;U^{\lhd};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u;B^{\smile}\\
&= \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;\lambda_1^{\smile};\lambda_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;\lambda_1^{\smile};\lambda_1;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \lambda_2;U^{\lhd};\mathsf{ran}\,V;\lambda_2^{\smile};\lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \lambda_2;U^{\lhd};\mathsf{ran}\,V;\lambda_2^{\smile};\lambda_2;U^{\lhd};\Psi_v^{\smile}\\
&= \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \lambda_2;(U^{\smile})^{\boxast};\mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \lambda_2;U^{\lhd};\mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \lambda_2;U^{\lhd};\mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile}\\
&\sqsubseteq \lambda_2;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \lambda_2;U^{\lhd};\Psi_v^{\smile}\\
&= \Psi_u;B^{\smile}
\end{aligned}
$$

This implies $\mathrm{X}_u;(B^{\smile})^{\boxast} = \mathrm{X}_u;B^{\smile}$ and $\Psi_u;(B^{\smile})^{\boxast} = \Psi_u;B^{\smile}$, and $\mathrm{X}_u;B^{\lhd} = \mathrm{X}_u \sqcup \mathrm{X}_u;B^{\smile};B$ and $\Psi_u;B^{\lhd} = \Psi_u \sqcup \Psi_u;B^{\smile};B$.



For $\mathrm{X}_v$ and $\Psi_v$, the fourth is contained in the second:

$$
\begin{aligned}
\mathrm{X}_v^{\smile};B;B^{\smile};B;B^{\smile} &= \mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile};B;B^{\smile} \sqcup \mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile};B;B^{\smile}\\
&= \mathsf{dom}\,V;c_1;U^{\rhd};\mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \mathsf{dom}\,V;c_1;U^{\rhd};\mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \mathsf{dom}\,V;U^{\boxast};\mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \mathsf{dom}\,V;U^{\boxast};\mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile}\\
&\sqsubseteq \mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile}\\
&= \mathrm{X}_v^{\smile};B;B^{\smile}\\
\Psi_v^{\smile};B;B^{\smile};B;B^{\smile} &= \mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile};B;B^{\smile} \sqcup \mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile};B;B^{\smile}\\
&= \mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile} \sqcup \mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile} \sqcup\\
&\quad \mathsf{ran}\,V;c_2;U^{\lhd};\mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \mathsf{ran}\,V;c_2;U^{\lhd};\mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile}\\
&\sqsubseteq \mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile} \sqcup \mathsf{ran}\,V;c_2;U^{\lhd};\Psi_v^{\smile}\\
&= \Psi_v^{\smile};B;B^{\smile}
\end{aligned}
$$

So we have $\Psi_v^{\smile};B^{\rhd} = \Psi_v^{\smile} \sqcup \Psi_v^{\smile};B;B^{\smile}$ and $\mathrm{X}_v^{\smile};B^{\rhd} = \mathrm{X}_v^{\smile} \sqcup \mathrm{X}_v^{\smile};B;B^{\smile}$.

These facts help to simplify the different components arising from the left-hand sides of the glued tabulation conditions:

$$
\begin{aligned}
\lambda_1^{\smile};\mathrm{X}_u;B^{\lhd};\mathrm{X}_u^{\smile};\lambda_1 &= \lambda_1^{\smile};\mathrm{X}_u;\mathrm{X}_u^{\smile};\lambda_1 \sqcup \lambda_1^{\smile};\mathrm{X}_u;B^{\smile};B;\mathrm{X}_u^{\smile};\lambda_1\\
&= c_1;U^{\rhd} \sqcup c_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u;\mathrm{X}_u^{\smile};\lambda_1 \sqcup U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u;\mathrm{X}_u^{\smile};\lambda_1\\
&= c_1;U^{\rhd}\\
\lambda_1^{\smile};\mathrm{X}_u;B^{\lhd};\Psi_u^{\smile};\lambda_2 &= \lambda_1^{\smile};\mathrm{X}_u;\Psi_u^{\smile};\lambda_2 \sqcup \lambda_1^{\smile};\mathrm{X}_u;B^{\smile};B;\Psi_u^{\smile};\lambda_2\\
&= \lambda_1^{\smile};\lambda_1;U^{\boxast};\lambda_2^{\smile};\lambda_2 \sqcup c_1;U^{\rhd};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u;\Psi_u^{\smile};\lambda_2 \sqcup U^{\boxast};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u;\Psi_u^{\smile};\lambda_2\\
&= U^{\boxast} \sqcup c_1;U^{\rhd};\mathsf{dom}\,V;U^{\boxast} \sqcup U^{\boxast};\mathsf{ran}\,V;U^{\lhd}\\
&= U^{\boxast}\\
\lambda_1^{\smile};\mathrm{X}_u;(B^{\smile})^{\boxast};\mathrm{X}_v &= \lambda_1^{\smile};\mathrm{X}_u;B^{\smile};\mathrm{X}_v \;=\; c_1;U^{\rhd};\mathrm{X}_v^{\smile};\mathrm{X}_v \sqcup U^{\boxast};\Psi_v^{\smile};\mathrm{X}_v\\
&= c_1;U^{\rhd};\mathsf{dom}\,V \sqcup U^{\boxast};V^{\smile} \;=\; U^{\boxast};V^{\smile}\\
\lambda_1^{\smile};\mathrm{X}_u;(B^{\smile})^{\boxast};\Psi_v &= \lambda_1^{\smile};\mathrm{X}_u;B^{\smile};\Psi_v \;=\; c_1;U^{\rhd};\mathrm{X}_v^{\smile};\Psi_v \sqcup U^{\boxast};\Psi_v^{\smile};\Psi_v\\
&= c_1;U^{\rhd};V \sqcup U^{\boxast};\mathsf{ran}\,V \;=\; U^{\boxast};\mathsf{ran}\,V\\
\mathrm{X}_v^{\smile};B^{\boxast};\Psi_u^{\smile};\lambda_2 &= \mathrm{X}_v^{\smile};B;\Psi_u^{\smile};\lambda_2 \;=\; \mathrm{X}_v^{\smile};\mathrm{X}_v;U^{\boxast} \sqcup \mathrm{X}_v^{\smile};\Psi_v;U^{\lhd};c_2\\
&= \mathsf{dom}\,V;U^{\boxast} \sqcup V;U^{\lhd};c_2 \;=\; \mathsf{dom}\,V;U^{\boxast}\\
\lambda_2^{\smile};\Psi_u;(B^{\smile})^{\boxast};\Psi_v &= \lambda_2^{\smile};\Psi_u;B^{\smile};\Psi_v \;=\; (U^{\smile})^{\boxast};\mathrm{X}_v^{\smile};\Psi_v \sqcup c_2;U^{\lhd};\Psi_v^{\smile};\Psi_v\\
&= (U^{\smile})^{\boxast};V \sqcup c_2;U^{\lhd};\mathsf{ran}\,V \;=\; c_2;U^{\lhd};\mathsf{ran}\,V\\
\lambda_2^{\smile};\Psi_u;B^{\lhd};\Psi_u^{\smile};\lambda_2 &= \lambda_2^{\smile};\Psi_u;\Psi_u^{\smile};\lambda_2 \sqcup \lambda_2^{\smile};\Psi_u;B^{\smile};B;\Psi_u^{\smile};\lambda_2\\
&= c_2;U^{\lhd} \sqcup (U^{\smile})^{\boxast};\mathsf{dom}\,V;\lambda_1^{\smile};\mathrm{X}_u;\Psi_u^{\smile};\lambda_2 \sqcup c_2;U^{\lhd};\mathsf{ran}\,V;\lambda_2^{\smile};\Psi_u;\Psi_u^{\smile};\lambda_2\\
&= c_2;U^{\lhd} \sqcup (U^{\smile})^{\boxast};\mathsf{dom}\,V;U^{\boxast} \sqcup c_2;U^{\lhd};\mathsf{ran}\,V;U^{\lhd};c_2\\
&= c_2;U^{\lhd}\\
\mathrm{X}_v^{\smile};B^{\rhd};\mathrm{X}_v &= \mathrm{X}_v^{\smile};\mathrm{X}_v \sqcup \mathrm{X}_v^{\smile};B;B^{\smile};\mathrm{X}_v\\
&= \mathsf{dom}\,V \sqcup \mathsf{dom}\,V;c_1;U^{\rhd};\mathrm{X}_v^{\smile};\mathrm{X}_v \sqcup \mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile};\mathrm{X}_v\\
&= \mathsf{dom}\,V \sqcup \mathsf{dom}\,V;c_1;U^{\rhd};\mathsf{dom}\,V \sqcup \mathsf{dom}\,V;U^{\boxast};V^{\smile}\\
&= \mathsf{dom}\,V \sqcup \mathsf{dom}\,V;c_1;U^{\rhd};\mathsf{dom}\,V
\end{aligned}
$$

$$\begin{aligned}
\mathrm{X}_v^{\smile};B^{\triangleright};\Psi_v &= \mathrm{X}_v^{\smile};\Psi_v \sqcup \mathrm{X}_v^{\smile};B;B^{\smile};\Psi_v\\
&= V \sqcup \mathsf{dom}\,V;c_1;U^{\triangleright};\mathrm{X}_v^{\smile};\Psi_v \sqcup \mathsf{dom}\,V;U^{\boxast};\Psi_v^{\smile};\Psi_v\\
&= V \sqcup \mathsf{dom}\,V;c_1;U^{\triangleright};V \sqcup \mathsf{dom}\,V;U^{\boxast};\mathsf{ran}\,V\\
&= V \sqcup \mathsf{dom}\,V;U^{\boxast};\mathsf{ran}\,V\\
\Psi_v^{\smile};B^{\triangleright};\Psi_v &= \Psi_v^{\smile};\Psi_v \sqcup \Psi_v^{\smile};B;B^{\smile};\Psi_v\\
&= \mathsf{ran}\,V \sqcup \mathsf{ran}\,V;(U^{\smile})^{\boxast};\mathrm{X}_v^{\smile};\Psi_v \sqcup \mathsf{ran}\,V;c_2;U^{\triangleleft};\Psi_v^{\smile};\Psi_v\\
&= \mathsf{ran}\,V \sqcup \mathsf{ran}\,V;(U^{\smile})^{\boxast};V \sqcup \mathsf{ran}\,V;c_2;U^{\triangleleft};\mathsf{ran}\,V\\
&= \mathsf{ran}\,V \sqcup \mathsf{ran}\,V;c_2;U^{\triangleleft};\mathsf{ran}\,V
\end{aligned}$$

This allows us to show the first three glued tabulation properties:

$$\begin{aligned}
\mathrm{X};\Psi^{\smile} &= (\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \mathrm{X}_v^{\smile};\Lambda_v);(\Lambda_u^{\smile};\Psi_u^{\smile};\lambda_2 \sqcup \Lambda_v^{\smile};\Psi_v)\\
&= \lambda_1^{\smile};\mathrm{X}_u;\Lambda_u;\Lambda_u^{\smile};\Psi_u^{\smile};\lambda_2 \sqcup \lambda_1^{\smile};\mathrm{X}_u;\Lambda_u;\Lambda_v^{\smile};\Psi_v \sqcup \mathrm{X}_v^{\smile};\Lambda_v;\Lambda_u^{\smile};\Psi_u^{\smile};\lambda_2 \sqcup \mathrm{X}_v^{\smile};\Lambda_v;\Lambda_v^{\smile};\Psi_v\\
&= \lambda_1^{\smile};\mathrm{X}_u;B^{\triangleleft};\Psi_u^{\smile};\lambda_2 \sqcup \lambda_1^{\smile};\mathrm{X}_u;(B^{\smile})^{\boxast};\Psi_v \sqcup \mathrm{X}_v^{\smile};B^{\boxast};\Psi_u^{\smile};\lambda_2 \sqcup \mathrm{X}_v^{\smile};B^{\triangleright};\Psi_v\\
&= U^{\boxast} \sqcup U^{\boxast};\mathsf{ran}\,V \sqcup \mathsf{dom}\,V;U^{\boxast} \sqcup V \sqcup \mathsf{dom}\,V;U^{\boxast};\mathsf{ran}\,V\\
&= U^{\boxast} \sqcup V\\
\mathrm{X};\mathrm{X}^{\smile} &= (\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \mathrm{X}_v^{\smile};\Lambda_v);(\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \Lambda_v^{\smile};\mathrm{X}_v)\\
&= \lambda_1^{\smile};\mathrm{X}_u;\Lambda_u;\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \lambda_1^{\smile};\mathrm{X}_u;\Lambda_u;\Lambda_v^{\smile};\mathrm{X}_v \sqcup \mathrm{X}_v^{\smile};\Lambda_v;\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \mathrm{X}_v^{\smile};\Lambda_v;\Lambda_v^{\smile};\mathrm{X}_v\\
&= \lambda_1^{\smile};\mathrm{X}_u;B^{\triangleleft};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \lambda_1^{\smile};\mathrm{X}_u;(B^{\smile})^{\boxast};\mathrm{X}_v \sqcup \mathrm{X}_v^{\smile};B^{\boxast};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \mathrm{X}_v^{\smile};B^{\triangleright};\mathrm{X}_v\\
&= c_1;U^{\triangleright} \sqcup U^{\boxast};V^{\smile} \sqcup V;U^{\boxast} \sqcup \mathsf{dom}\,V \sqcup \mathsf{dom}\,V;c_1;U^{\triangleright};\mathsf{dom}\,V\\
&= \mathsf{dom}\,V \sqcup c_1;U^{\triangleright};c_1
\end{aligned}$$

$$\begin{aligned}
\Psi;\Psi^{\smile} &= (\lambda_2^{\smile};\Psi_u;\Lambda_u \sqcup \Psi_v^{\smile};\Lambda_v);(\Lambda_u^{\smile};\Psi_u^{\smile};\lambda_2 \sqcup \Lambda_v^{\smile};\Psi_v)\\
&= \lambda_2^{\smile};\Psi_u;\Lambda_u;\Lambda_u^{\smile};\Psi_u^{\smile};\lambda_2 \sqcup \lambda_2^{\smile};\Psi_u;\Lambda_u;\Lambda_v^{\smile};\Psi_v \sqcup \Psi_v^{\smile};\Lambda_v;\Lambda_u^{\smile};\Psi_u^{\smile};\lambda_2 \sqcup \Psi_v^{\smile};\Lambda_v;\Lambda_v^{\smile};\Psi_v\\
&= \lambda_2^{\smile};\Psi_u;B^{\triangleleft};\Psi_u^{\smile};\lambda_2 \sqcup \lambda_2^{\smile};\Psi_u;(B^{\smile})^{\boxast};\Psi_v \sqcup \Psi_v^{\smile};B^{\boxast};\Psi_u^{\smile};\lambda_2 \sqcup \Psi_v^{\smile};B^{\triangleright};\Psi_v\\
&= c_2;U^{\triangleleft} \sqcup c_2;U^{\triangleleft};\mathsf{ran}\,V \sqcup \mathsf{ran}\,V;U^{\triangleleft};c_2 \sqcup \mathsf{ran}\,V \sqcup \mathsf{ran}\,V;c_2;U^{\triangleleft};\mathsf{ran}\,V\\
&= \mathsf{ran}\,V \sqcup c_2;U^{\triangleleft};c_2
\end{aligned}$$

We now prepare to show the last property, namely:

$$\mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) = \mathbb{I} \ .$$

With Lemma A.1.2.iii), we have:

$$\begin{aligned}
\Lambda_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqsubseteq c_3 \quad&\Leftrightarrow\quad \mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqsubseteq \Lambda_v;\Lambda_u^{\smile} \;\Leftrightarrow\; \mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqsubseteq B^{\boxast}\\
\Lambda_v^{\smile};\mathrm{X}_v;c_1;\mathrm{X}_v^{\smile};\Lambda_v \sqsubseteq \mathbb{I} \quad&\Leftrightarrow\quad \mathrm{X}_v;c_1;\mathrm{X}_v^{\smile} \sqsubseteq \Lambda_v;\Lambda_v^{\smile} \;\Leftrightarrow\; \mathrm{X}_v;c_1;\mathrm{X}_v^{\smile} \sqsubseteq B^{\triangleright}
\end{aligned}$$

The last inclusion of the first row holds true by definition of $B$; the last inclusion of the second row is shown as follows:

$$\mathrm{X}_v;c_1;\mathrm{X}_v^{\smile} \;=\; \mathrm{X}_v;\lambda_1^{\smile};\lambda_1;\mathrm{X}_v^{\smile} = \mathrm{X}_v;\lambda_1^{\smile};\lambda_1;\lambda_1^{\smile};\lambda_1;\mathrm{X}_v^{\smile} \;\sqsubseteq\; \mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u;\mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile} \sqsubseteq B;B^{\smile} \sqsubseteq B^{\triangleright}$$

With these preparations, we get:

$$\begin{aligned}
\mathrm{X}^{\smile};c_1;\mathrm{X} &= (\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \Lambda_v^{\smile};\mathrm{X}_v);c_1;(\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \mathrm{X}_v^{\smile};\Lambda_v)\\
&= \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile};\Lambda_v \sqcup \Lambda_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_v^{\smile};\mathrm{X}_v;c_1;\mathrm{X}_v^{\smile};\Lambda_v\\
&= \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile};\Lambda_v \sqcup \Lambda_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_v^{\smile};\mathrm{X}_v;c_1;\mathrm{X}_v^{\smile};\Lambda_v
\end{aligned}$$



The corresponding inclusions for $\Psi$ are shown in the same way, so we have the following two-sided approximations for $\mathrm{X}^{\smile};c_1;\mathrm{X}$ and $\Psi^{\smile};c_2;\Psi$:

$$\begin{aligned}
\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\mathrm{X}_u;\Lambda_u \;&\sqsubseteq\; \mathrm{X}^{\smile};c_1;\mathrm{X} \;\sqsubseteq\; \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \mathbb{I} \ ,\\
\Lambda_u^{\smile};\Psi_u^{\smile};\Psi_u;\Lambda_u \;&\sqsubseteq\; \Psi^{\smile};c_2;\Psi \;\sqsubseteq\; \Lambda_u^{\smile};\Psi_u^{\smile};\Psi_u;\Lambda_u \sqcup \mathbb{I} \ .
\end{aligned}$$

Taking the join, this yields:

$$\begin{aligned}
c_3 &= \Lambda_u^{\smile};\Lambda_u \;=\; \Lambda_u^{\smile};(\mathrm{X}_u^{\smile};\mathrm{X}_u \sqcup \Psi_u^{\smile};\Psi_u);\Lambda_u \;=\; \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_u^{\smile};\Psi_u^{\smile};\Psi_u;\Lambda_u\\
&\sqsubseteq \mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi \;\sqsubseteq\; \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_u^{\smile};\Psi_u^{\smile};\Psi_u;\Lambda_u \sqcup \mathbb{I} \;=\; \mathbb{I}
\end{aligned}$$

Now let us consider the parameter components. We have:

$$\begin{aligned}
\mathrm{X}^{\smile};v_1;\mathrm{X} &= (\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1 \sqcup \Lambda_v^{\smile};\mathrm{X}_v);v_1;(\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \mathrm{X}_v^{\smile};\Lambda_v)\\
&= \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1;v_1;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile};\Lambda_v \sqcup \Lambda_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Lambda_v^{\smile};\mathrm{X}_v;\mathrm{X}_v^{\smile};\Lambda_v
\end{aligned}$$

The first join component of this is contained in $c_3$:

$$\begin{aligned}
\Lambda_u^{\smile};\mathrm{X}_u^{\smile};\lambda_1;v_1;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqsubseteq c_3 \quad&\Leftrightarrow\quad \mathrm{X}_u^{\smile};\lambda_1;v_1;\lambda_1^{\smile};\mathrm{X}_u \sqsubseteq \Lambda_u;\Lambda_u^{\smile}\\
&\Leftrightarrow\quad \mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqsubseteq B^{\triangleleft}\\
&\Leftarrow\quad \mathrm{X}_u^{\smile};\lambda_1;\mathrm{X}_v^{\smile};\mathrm{X}_v;\lambda_1^{\smile};\mathrm{X}_u \sqsubseteq B^{\smile};B
\end{aligned}$$

With these preparations, we then have (the inclusions for $\Psi$ are obtained in the same way):

$$\begin{aligned}
\Lambda_v^{\smile};\mathrm{X}_v;\mathrm{X}_v^{\smile};\Lambda_v \;&\sqsubseteq\; \mathrm{X}^{\smile};v_1;\mathrm{X} \;\sqsubseteq\; c_3 \sqcup \Lambda_v^{\smile};\mathrm{X}_v;\mathrm{X}_v^{\smile};\Lambda_v\\
\Lambda_v^{\smile};\Psi_v;\Psi_v^{\smile};\Lambda_v \;&\sqsubseteq\; \Psi^{\smile};v_2;\Psi \;\sqsubseteq\; c_3 \sqcup \Lambda_v^{\smile};\Psi_v;\Psi_v^{\smile};\Lambda_v
\end{aligned}$$

From this, we may form the intersection and obtain:

$$\begin{aligned}
v_3 &= \Lambda_v^{\smile};\Lambda_v \;=\; \Lambda_v^{\smile};(\mathrm{X}_v;\mathrm{X}_v^{\smile} \sqcap \Psi_v;\Psi_v^{\smile});\Lambda_v \;=\; \Lambda_v^{\smile};\mathrm{X}_v;\mathrm{X}_v^{\smile};\Lambda_v \sqcap \Lambda_v^{\smile};\Psi_v;\Psi_v^{\smile};\Lambda_v\\
&\sqsubseteq \mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi \;\sqsubseteq\; c_3 \sqcup (\Lambda_v^{\smile};\mathrm{X}_v;\mathrm{X}_v^{\smile};\Lambda_v \sqcap \Lambda_v^{\smile};\Psi_v;\Psi_v^{\smile};\Lambda_v) \;\sqsubseteq\; c_3 \sqcup v_3
\end{aligned}$$

Now we may take both components together:

$$\mathbb{I} = c_3 \sqcup v_3 \sqsubseteq \mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \sqsubseteq \mathbb{I}$$

This completes the proof of the glued tabulation properties. □

**Proof of commutativity of pullouts (Theorem 6.2.7)**: The pullout construction gives us:

$$\Phi;\mathrm{X} = \Xi;\Psi \qquad\Leftrightarrow\qquad \Phi;\lambda_1^{\smile};\mathrm{X}_u;\Lambda_u \sqcup \Phi;\mathrm{X}_v^{\smile};\Lambda_v = \Xi;\lambda_2^{\smile};\Psi_u;\Lambda_u \sqcup \Xi;\Psi_v^{\smile};\Lambda_v$$

We only show one inclusion; the opposite inclusion follows in the same way.

$$
\begin{array}{rll}
\Phi{;}\lambda_1^{\smile}{;}\mathrm{X}_u{;}\Lambda_u \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}\Lambda_u & \Leftrightarrow\ \Phi{;}\lambda_1^{\smile}{;}\mathrm{X}_u \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}\Lambda_u{;}\Lambda_u^{\smile} & \\
& \Leftrightarrow\ \Phi{;}\lambda_1^{\smile}{;}\mathrm{X}_u \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}\Lambda_u{;}\Lambda_u^{\smile} & \\
& \Leftrightarrow\ \Phi{;}\lambda_1^{\smile}{;}\mathrm{X}_u \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}B^{\lhd} & \\
& \Leftarrow\ \Phi{;}\lambda_1^{\smile}{;}\mathrm{X}_u \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u & \\
& \Leftrightarrow\ \Phi{;}\lambda_1^{\smile} \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}\mathrm{X}_u^{\smile} & \\
& \Leftrightarrow\ \Phi{;}\lambda_1^{\smile} \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\lambda_2{;}(U^{\smile})^{\boxast}{;}\lambda_1^{\smile} & \\
& \Leftrightarrow\ \Phi{;}\lambda_1^{\smile} \sqsubseteq \Xi{;}c_2{;}(U^{\smile})^{\boxast}{;}\lambda_1^{\smile} & \\
& \Leftrightarrow\ u_0{;}\Phi{;}\lambda_1^{\smile} \sqsubseteq u_0{;}\Xi{;}c_2{;}(\Xi{;}u_0{;}\Phi^{\smile})^{\boxast}{;}\lambda_1^{\smile} & \text{interf. pres.} \\
& \Leftrightarrow\ u_0{;}\Phi{;}\lambda_1^{\smile} \sqsubseteq u_0{;}\Xi{;}(\Xi{;}u_0{;}\Phi^{\smile})^{\boxast}{;}\lambda_1^{\smile} & \text{Def. } c_2 \\
& \Leftarrow\ u_0 \sqsubseteq \mathsf{dom}\,\Xi &
\end{array}
$$

For the variable component, we have to be rather careful, since we do not have full injectivity of $v_0{;}\Phi$, but only almost-injectivity. We first show the inclusion of the $v_0$-component of the variable part in the right-hand side's variable part:

$$
\begin{array}{lll}
& v_0{;}\Phi{;}\mathrm{X}_v^{\smile}{;}\Lambda_v \sqsubseteq \Xi{;}\Psi_v^{\smile}{;}\Lambda_v & \\
\Leftrightarrow & v_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\Psi_v^{\smile}{;}\Lambda_v{;}\Lambda_v^{\smile} & \\
\Leftrightarrow & v_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\Psi_v^{\smile}{;}B^{\rhd} & \\
\Leftrightarrow & v_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\Psi_v^{\smile} \sqcup \Xi{;}\Psi_v^{\smile}{;}B{;}B^{\smile} & \\
\Leftrightarrow & v_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\Psi_v^{\smile} \sqcup \Xi{;}\mathsf{ran}\,V{;}(U^{\smile})^{\boxast}{;}\mathrm{X}_v^{\smile} \sqcup \Xi{;}\mathsf{ran}\,V{;}c_2{;}U^{\lhd}{;}\Psi_v^{\smile} & \\
\Leftarrow & v_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\Psi_v^{\smile} \sqcup \Xi{;}\mathsf{ran}\,V{;}c_2{;}U^{\lhd}{;}\Psi_v^{\smile} & \\
\Leftrightarrow & v_0{;}\Phi{;}\mathrm{X}_v^{\smile}{;}\Psi_v \sqsubseteq \Xi \sqcup \Xi{;}\mathsf{ran}\,V{;}c_2{;}U^{\lhd} & \\
\Leftrightarrow & v_0{;}\Phi{;}V \sqsubseteq \Xi \sqcup \Xi{;}\mathsf{ran}\,V{;}c_2{;}U^{\lhd} & \\
\Leftrightarrow & v_0{;}\Phi{;}\Phi^{\smile}{;}v_0{;}\Xi \sqsubseteq \Xi \sqcup \Xi{;}v_2{;}c_2{;}(\Phi^{\smile}{;}u_0{;}\Xi)^{\lhd} & \\
\Leftarrow & u_0{;}v_0{;}\Phi{;}\Phi^{\smile}{;}v_0{;}u_0{;}\Xi \sqsubseteq \Xi{;}v_2{;}c_2{;}(\Xi^{\smile}{;}u_0{;}\Phi{;}\Phi^{\smile}{;}u_0{;}\Xi)^{*} & \Phi\ \text{almost-inj. bes. } u_0 \\
\Leftarrow & u_0{;}v_0 \sqsubseteq \mathsf{dom}\,(\Xi{;}v_2{;}c_2) &
\end{array}
$$

To finish, it is sufficient to show inclusion of the $u_0$-component in the right-hand side's interface part:

$$
\begin{array}{rl}
u_0{;}\Phi{;}\mathrm{X}_v^{\smile}{;}\Lambda_v \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}\Lambda_u & \Leftrightarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}\Lambda_u{;}\Lambda_v^{\smile} \\
& \Leftrightarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}(B^{\smile})^{\boxast} \\
& \Leftrightarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\Psi_u{;}B^{\smile} \\
& \Leftrightarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}\lambda_2^{\smile}{;}\lambda_2{;}(U^{\smile})^{\boxast}{;}\mathrm{X}_v^{\smile} \sqcup \Xi{;}\lambda_2^{\smile}{;}\lambda_2{;}U^{\lhd}{;}\Psi_v^{\smile} \\
& \Leftrightarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}c_2{;}(U^{\smile})^{\boxast}{;}\mathrm{X}_v^{\smile} \sqcup \Xi{;}c_2{;}U^{\lhd}{;}\Psi_v^{\smile} \\
& \Leftarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}c_2{;}U^{\smile}{;}\mathrm{X}_v^{\smile} \\
& \Leftrightarrow\ u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \sqsubseteq \Xi{;}c_2{;}\Xi^{\smile}{;}u_0{;}\Phi{;}\mathrm{X}_v^{\smile} \\
& \Leftarrow\ u_0 \sqsubseteq \mathsf{dom}\,(\Xi{;}c_2)
\end{array}
$$

□



## B.2 Correctness of the Pullout Complement Construction

**Proof of Theorem 6.3.1**: We have:

$$
\begin{array}{rcll}
\Phi^{\smile}{;}v_0{;}\Xi &=& \Phi^{\smile}{;}v_0{;}\Phi{;}\mathrm{X}{;}\Psi^{\smile} \ =\ v_1{;}\mathrm{X}{;}\Psi^{\smile} & \\
\Phi^{\smile}{;}u_0{;}\Xi &=& \Phi^{\smile}{;}u_0{;}\Phi{;}\mathrm{X}{;}\Psi^{\smile} \ =\ u_1{;}\mathrm{X}{;}\Psi^{\smile} & \\
\Phi^{\smile}{;}u_0{;}\Xi{;}\Xi^{\smile}{;}u_0{;}\Phi &=& u_1{;}\mathrm{X}{;}\Psi^{\smile}{;}\Psi{;}\mathrm{X}^{\smile}{;}u_1 \ =\ u_1{;}\mathrm{X}{;}\mathrm{X}^{\smile}{;}u_1 & \mathsf{ran}\,(u_1{;}\mathrm{X}) \sqsubseteq \mathsf{ran}\,\Psi \\
\Xi^{\smile}{;}u_0{;}\Phi{;}\Phi^{\smile}{;}u_0{;}\Xi &=& \Psi{;}\mathrm{X}^{\smile}{;}u_1{;}\mathrm{X}{;}\Psi^{\smile} \ \sqsubseteq\ \Psi{;}\Psi^{\smile} \ =\ \mathbb{I} & u_1{;}\mathrm{X}\ \text{univalent}
\end{array}
$$

For the parameter component, we additionally have the following:

$$
\begin{array}{rcl}
\mathsf{dom}\,(\Phi^{\smile}{;}v_0{;}\Xi) &=& \mathsf{dom}\,(v_1{;}\mathrm{X}{;}\Psi^{\smile}) \ =\ \mathsf{dom}\,(v_1{;}\mathrm{X}) \\
\mathsf{ran}\,(\Phi^{\smile}{;}v_0{;}\Xi) &=& \mathsf{ran}\,(\Phi^{\smile}{;}v_0{;}\Phi{;}\mathrm{X}{;}\Psi^{\smile}) \ =\ \mathsf{ran}\,(v_1{;}\mathrm{X}{;}\Psi^{\smile}) \ =\ \mathsf{ran}\,(v_3{;}\Psi^{\smile})
\end{array}
$$

The partitioning of $G_2$ into parameter part $v_2$ and non-parameter part $c_2$ can be copied from $G_3$:

$$
\begin{array}{rcl}
c_2 &:=& \mathsf{ran}\,(c_3{;}\Psi^{\smile}) \\
v_2 &:=& \mathsf{ran}\,(v_3{;}\Psi^{\smile}) \ =\ \mathsf{ran}\,(v_0{;}\Phi{;}\mathrm{X}{;}\Psi^{\smile}) \ =\ \mathsf{ran}\,(v_0{;}\Xi) \\
v_3 &=& \mathsf{ran}\,(v_0{;}\Phi{;}\mathrm{X}) \ =\ \mathsf{ran}\,(v_0{;}\Phi{;}\mathrm{X}{;}\Psi^{\smile}{;}\Psi) \ =\ \mathsf{ran}\,(v_0{;}\Xi{;}\Psi) \ =\ \mathsf{ran}\,(v_2{;}\Psi) \\
c_2 \sqcup v_2 &=& \mathsf{ran}\,(c_3{;}\Psi^{\smile}) \sqcup \mathsf{ran}\,(v_3{;}\Psi^{\smile}) \ =\ \mathsf{ran}\,((c_3 \sqcup v_3){;}\Psi^{\smile}) \ =\ \mathsf{ran}\,(\Psi^{\smile}) \ =\ \mathbb{I}
\end{array}
$$

With all this, we can derive the four pullout conditions:

$$
\begin{array}{rcll}
\mathrm{X}{;}\mathrm{X}^{\smile} &=& c_1 \sqcup u_1{;}\mathrm{X}{;}\mathrm{X}^{\smile}{;}u_1 \sqcup v_1{;}\mathsf{dom}\,\mathrm{X} & \\
&=& c_1 \sqcup (u_1{;}\mathrm{X}{;}\mathrm{X}^{\smile}{;}u_1)^{+} \sqcup \mathsf{dom}\,(v_1{;}\mathrm{X}) & \text{(ii)} \\
&=& c_1 \sqcup (\Phi^{\smile}{;}u_0{;}\Xi{;}\Xi^{\smile}{;}u_0{;}\Phi)^{+} \sqcup \mathsf{dom}\,(\Phi^{\smile}{;}v_0{;}\Xi) & \\
&=& c_1{;}(\Phi^{\smile}{;}u_0{;}\Xi)^{\rhd}{;}c_1 \sqcup \mathsf{dom}\,(\Phi^{\smile}{;}v_0{;}\Xi) & \\
\Psi{;}\Psi^{\smile} &=& \mathbb{I} \ =\ c_2 \sqcup \mathsf{ran}\,(\Phi^{\smile}{;}v_0{;}\Xi) \ =\ c_2{;}(\Phi^{\smile}{;}u_0{;}\Xi)^{\lhd}{;}c_2 \sqcup \mathsf{ran}\,(\Phi^{\smile}{;}v_0{;}\Xi) & \\
\mathrm{X}{;}\Psi^{\smile} &=& \mathsf{ran}\,\Phi{;}\mathrm{X}{;}\Psi^{\smile} \ =\ u_1{;}\mathrm{X}{;}\Psi^{\smile} \sqcup v_1{;}\mathrm{X}{;}\Psi^{\smile} & \text{5.4.7.vi)} \\
&=& \Phi^{\smile}{;}u_0{;}\Xi \sqcup \Phi^{\smile}{;}v_0{;}\Xi \ =\ (\Phi^{\smile}{;}u_0{;}\Xi)^{\boxast} \sqcup \Phi^{\smile}{;}v_0{;}\Xi &
\end{array}
$$

$$
\begin{array}{rcll}
\mathbb{I} &=& \mathsf{ran}\,\mathrm{X} \sqcup \mathsf{ran}\,\Psi & \text{(iii), (iv) imply gluing cond.} \\
&=& \mathsf{ran}\,(c_1{;}\mathrm{X}) \sqcup \mathsf{ran}\,(v_1{;}\mathrm{X}) \sqcup \mathsf{ran}\,(c_2{;}\Psi) \sqcup \mathsf{ran}\,(v_2{;}\Psi) & \\
&=& \mathsf{ran}\,(c_1{;}\mathrm{X}) \sqcup \mathsf{ran}\,(c_2{;}\Psi) \sqcup (\mathsf{ran}\,(v_1{;}\mathrm{X}) \sqcap \mathsf{ran}\,(v_2{;}\Psi)) & \mathsf{ran}\,(v_1{;}\mathrm{X}) = v_3 = \mathsf{ran}\,(v_2{;}\Psi) \\
&=& \mathsf{ran}\,(c_1{;}\mathrm{X}) \sqcup \mathsf{ran}\,(c_2{;}\Psi) \sqcup (\mathrm{X}^{\smile}{;}v_1{;}\mathrm{X} \sqcap \mathsf{ran}\,(v_2{;}\Psi)) & \mathsf{ran}\,(v_2{;}\Psi) \sqsubseteq \mathsf{ran}\,(v_1{;}\mathrm{X}) \\
&=& \mathrm{X}^{\smile}{;}c_1{;}\mathrm{X} \sqcup \Psi^{\smile}{;}c_2{;}\Psi \sqcup (\mathrm{X}^{\smile}{;}v_1{;}\mathrm{X} \sqcap \Psi^{\smile}{;}v_2{;}\Psi) & c_1{;}\mathrm{X}\ \text{and}\ \Psi\ \text{univalent}
\end{array}
$$

For showing that $\Xi$ is a standard gluing morphism, we first show interface preservation of $\Xi$, and we start with calculating $v_2^{\sim}$:

$$
\begin{array}{rcll}
v_2^{\sim} &=& (\mathsf{ran}\,(v_3{;}\Psi^{\smile}))^{\sim} & \\
&=& \mathsf{ran}\,(((\mathsf{ran}\,\Psi) \setminus v_3){;}\Psi^{\smile}) \sqcup (\mathsf{dom}\,\Psi)^{\sim} & \text{Lemma A.4.2.ii)} \\
&=& \mathsf{ran}\,(((\mathsf{ran}\,\Psi) \setminus v_3){;}\Psi^{\smile}) & \Psi\ \text{total} \\
&=& \Psi{;}((\mathsf{ran}\,\Psi) \setminus v_3){;}\Psi^{\smile} & \Psi\ \text{univalent}
\end{array}
$$

Furthermore, the inclusion

$$\begin{array}{rcll}
((\mathsf{ran}\,\Psi) \setminus v_3) & = & ((\mathsf{ran}\,\mathrm{X} \rightarrow \mathsf{ran}\,(\Phi;\mathrm{X})) \setminus v_3) & \text{Def. } \Psi \\
& \sqsubseteq & (((\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X})) \setminus v_3) & \text{Lemma 2.7.10} \\
& = & (((\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup (u_3 \sqcup v_3)) \setminus v_3) & \\
& \sqsubseteq & (\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup u_3 \ , &
\end{array}$$

together with (iv) and (vi) implies the following:

$$\mathrm{X};((\mathsf{ran}\,\Psi) \setminus v_3) \sqsubseteq u_1;\mathrm{X} \ . \qquad (*)$$

This allows us to obtain the second interface preservation condition:

$$\begin{array}{cll}
& \mathsf{dom}\,(\Xi;(v_2^{\sim} \sqcup u_2)) & \\
= & \mathsf{dom}\,(\Xi;\Psi;((\mathsf{ran}\,\Psi) \setminus v_3);\Psi^{\smile} \sqcup \Xi;\Psi;u_3;\Psi^{\smile}) & \\
= & \mathsf{dom}\,(\Phi;\mathrm{X};\Psi^{\smile};\Psi;((\mathsf{ran}\,\Psi) \setminus v_3);\Psi^{\smile} \sqcup \Phi;\mathrm{X};\Psi^{\smile};\Psi;u_3;\Psi^{\smile}) & \\
= & \mathsf{dom}\,(\Phi;\mathrm{X};((\mathsf{ran}\,\Psi) \setminus v_3) \sqcup \Phi;\mathrm{X};u_3) & u_3 \sqsubseteq \mathsf{ran}\,\Psi \\
= & \mathsf{dom}\,(\Phi;\mathrm{X};((\mathsf{ran}\,\Psi) \setminus v_3) \sqcup \Phi;u_1;\mathrm{X}) & \text{(vi), (ii)} \\
= & \mathsf{dom}\,(\Phi;u_1;\mathrm{X}) & (*) \\
= & \mathsf{dom}\,(\Phi;u_1) & \text{(ii)} \\
= & u_0 & \Phi \text{ interface pres.}
\end{array}$$

The first interface preservation condition follows from (v):

$$\begin{array}{rcl}
v_2 \sqcap v_2^{\sim} & = & \Psi;v_3;\Psi^{\smile} \sqcap \Psi;((\mathsf{ran}\,\Psi) \setminus v_3);\Psi^{\smile} \\
& = & \Psi;(v_3 \sqcap ((\mathsf{ran}\,\Psi) \setminus v_3));\Psi^{\smile} \ \sqsubseteq \ \Psi;(v_3 \sqcap v_3^{\sim});\Psi^{\smile} \ \sqsubseteq \ \Psi;u_3;\Psi^{\smile} \ = \ u_2
\end{array}$$

Univalence of $\Xi;b_2 = \Xi;(u_2 \sqcap v_2)$ follows from the second interface preservation condition together with univalence of $\Phi$ (i) and of $u_1;\mathrm{X}$ (ii).

Univalence of $u_0;\Xi$:

$$\Xi^{\smile};u_0;\Xi = \Psi;\mathrm{X}^{\smile};\Phi^{\smile};u_0;\Phi;\mathrm{X};\Psi^{\smile} = \Psi;\mathrm{X}^{\smile};u_1;\mathrm{X};\Psi^{\smile} = \Psi;u_2;\Psi^{\smile} = u_3$$

Totality of $\Xi$ on $u_0$:

$$\begin{array}{rcll}
\mathsf{dom}\,\Xi & = & \mathsf{dom}\,(\Phi;\mathrm{X};\Psi^{\smile}) & \\
& = & \mathsf{dom}\,(\Phi;\mathrm{X};\mathsf{ran}\,\Psi) & \\
& = & \mathsf{dom}\,(\Phi;\mathrm{X};(\mathsf{ran}\,\mathrm{X} \rightarrow \mathsf{ran}\,(\Phi;\mathrm{X}))) & \\
& \sqsupseteq & \mathsf{dom}\,(\Phi;\mathrm{X};(\mathsf{ran}\,\mathrm{X} \rightarrow \mathsf{ran}\,(\Phi;\mathrm{X}))) & \\
& \sqsupseteq & \mathsf{dom}\,(\Phi;\mathrm{X};\mathsf{ran}\,(\Phi;\mathrm{X})) & \text{Lemma 2.6.9} \\
& = & \mathsf{dom}\,(\Phi;\mathrm{X}) & \\
& = & \mathsf{dom}\,(\Phi;\mathsf{dom}\,\mathrm{X}) & \\
& \sqsupseteq & \mathsf{dom}\,(\Phi;u_1) & \text{(ii)} \\
& = & \mathsf{dom}\,(\Phi;\mathsf{ran}\,(u_0;\Phi)) & \\
& \sqsupseteq & \mathsf{dom}\,(u_0;\Phi) & \\
& = & u_0 & \text{Def. 6.2.1.ii)}
\end{array}$$



$\Xi$ almost-injective besides $u_0$:

$$\begin{array}{rcll}
\Xi;\Xi^{\smile} & = & \Phi;\mathrm{X};\Psi^{\smile};\Psi;\mathrm{X}^{\smile};\Phi^{\smile} & \\
& \sqsubseteq & \Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile} & \Psi \text{ univalent} \\
& \sqsubseteq & \Phi;(\mathbb{I} \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1);\Phi^{\smile} & \mathrm{X} \text{ alm. inj.} \\
& = & \Phi;\Phi^{\smile} \sqcup \Phi;u_1;\mathrm{X};\mathrm{X}^{\smile};u_1;\Phi^{\smile} & \\
& \sqsubseteq & \mathbb{I} \sqcup u_0;\Phi;\Phi^{\smile};u_0 \sqcup u_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};u_0 & \Phi \text{ alm. inj. and inf. pres.} \\
& = & \mathbb{I} \sqcup u_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};u_0 & \text{(ii)} \\
& = & \mathbb{I} \sqcup u_0;\Phi;\mathrm{X};\Psi^{\smile};\Psi;\mathrm{X}^{\smile};\Phi^{\smile};u_0 & \mathsf{ran}\,(\Phi;\mathrm{X}) \sqsubseteq \mathsf{ran}\,\Psi \\
& = & \mathbb{I} \sqcup u_0;\Xi;\Xi^{\smile};u_0 & \square
\end{array}$$

For showing the correctness of the general pullout complement construction, we assume throughout the remainder of this section the setup of the statement of Theorem 6.3.2, where the construction proceeded along the following diagram:

$$\begin{tikzcd}
\mathcal{L} \arrow[d, "\mathrm{X}"'] & \mathcal{G} \arrow[l, "\Phi"'] \arrow[d, "\Xi"] \\
\mathcal{A} \arrow[dr, "\nu"] & \mathcal{H} \arrow[l, "\Psi"'] \\
\mathcal{S} \arrow[u, "\lambda"] \arrow[r, "Z"] \arrow[ur, "\iota"] & \mathcal{V} \arrow[u, "\kappa"']
\end{tikzcd}$$

References (i) to (vii) refer to the preconditions of Theorem 6.3.2.

First of all, we show the analogon of Lemma 5.4.7.vi):

**Lemma B.2.1** [←182, 183] $\mathrm{X};c_3 \sqsubseteq u_1;\mathrm{X}$ , $\mathrm{X};v_3 = v_1;\mathrm{X} \sqcup u_1;\mathrm{X};v_3$ , $\mathrm{X};\Psi^{\smile} = \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile}$ .

**Proof**:

$$\begin{array}{rcll}
\mathrm{X};c_3 & = & \mathrm{X};(\mathsf{ran}\,\mathrm{X})^{\sim} \sqcup \mathrm{X};\mathsf{ran}\,(u_1;\mathrm{X}) & \\
& \sqsubseteq & u_1;\mathrm{X} \sqcup \mathrm{X};\mathrm{X}^{\smile};u_1;\mathrm{X} & \text{(iii)} \\
& \sqsubseteq & u_1;\mathrm{X} \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1;\mathrm{X} & \text{(ii)} \\
& \sqsubseteq & u_1;\mathrm{X} & \text{(i)} \\
\mathrm{X};v_3 & = & \mathrm{X};\mathsf{ran}\,(v_1;\mathrm{X}) & \\
& \sqsubseteq & \mathrm{X};\mathrm{X}^{\smile};v_1;\mathrm{X};v_3 & \\
& \sqsubseteq & v_1;\mathrm{X} \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};v_1;\mathrm{X};v_3 & \text{(ii)} \\
& \sqsubseteq & v_1;\mathrm{X} \sqcup u_1;\mathrm{X};v_3 & \text{(i)} \\
\mathrm{X};\Psi^{\smile} & = & \mathrm{X};\mathsf{ran}\,\Psi;\Psi^{\smile} \ = \ \mathrm{X};(c_3 \sqcup v_3);\Psi^{\smile} \ = \ \mathrm{X};c_3;\Psi^{\smile} \sqcup \mathrm{X};v_3;\Psi^{\smile} & \\
& = & u_1;\mathrm{X};c_3;\Psi^{\smile} \sqcup u_1;\mathrm{X};v_3;\Psi^{\smile} \sqcup v_1;\mathrm{X};v_3;\Psi^{\smile} & \\
& = & u_1;\mathrm{X};\Psi^{\smile} \sqcup v_1;\mathrm{X};\Psi^{\smile} \ = \ (u_1 \sqcup v_1);\mathrm{X};\Psi^{\smile} \ = \ \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} & \square
\end{array}$$

**Lemma B.2.2** [←183] $\mathsf{ran}\,\Psi = (\mathsf{ran}\,\mathrm{X})^{\smile} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X})$

**Proof:**
$$\begin{aligned}\mathsf{ran}\,\Psi &= \mathsf{ran}\,(\iota^{\smile};\lambda) \sqcup \mathsf{ran}\,(\kappa^{\smile};\nu^{\smile}) \;=\; \mathsf{ran}\,\lambda \sqcup \mathsf{ran}\,\nu^{\smile}\\ &= (\mathsf{ran}\,\mathrm{X})^{\smile} \sqcup u_3 \sqcup v_3 \;=\; (\mathsf{ran}\,\mathrm{X})^{\smile} \sqcup \mathsf{ran}\,(\Phi;\mathrm{X})\end{aligned}$$
□

**Lemma B.2.3** [←182] $c_3;\Theta = u_3;v_3$

**Proof**: Because of image preservation, we have:

$$c_3;\Theta = ((\mathsf{ran}\,\mathrm{X})^{\smile} \sqcup u_3);v_3;\Theta = u_3;v_3;\Theta = u_3;v_3$$
□

**Proof of Theorem 6.3.2**: $Z$ is univalent and injective:

$$\begin{aligned}Z^{\smile};Z &= \nu^{\smile};\lambda^{\smile};\lambda;\nu \;\sqsubseteq\; \nu^{\smile};\nu \;=\; \mathbb{I}\\ Z;Z^{\smile} &= \lambda;\nu;\nu^{\smile};\lambda^{\smile} \;=\; \lambda;\Theta;\lambda^{\smile} \;=\; \lambda;u_3;v_3;\lambda^{\smile} \;\sqsubseteq\; \mathbb{I} && \text{Lemma B.2.3}\end{aligned}$$

This implies the following simpler shapes for the gluing properties for $Z$:

$$\begin{aligned}\iota;\kappa^{\smile} &= Z^{\boxast} = Z\\ \iota;\iota^{\smile} &= Z^{\triangleright} = \mathbb{I} \sqcup Z;Z^{\smile} = \mathbb{I}\\ \kappa;\kappa^{\smile} &= Z^{\triangleleft} = \mathbb{I}\\ \iota^{\smile};\iota \sqcup \kappa^{\smile};\kappa &= \mathbb{I}\end{aligned}$$

Then we have:

$$\begin{aligned}\Psi^{\smile};\Psi &= (\lambda^{\smile};\iota \sqcup \nu;\kappa);(\iota^{\smile};\lambda \sqcup \kappa^{\smile};\nu^{\smile})\\ &= \lambda^{\smile};\iota;\iota^{\smile};\lambda \sqcup \lambda^{\smile};\iota;\kappa^{\smile};\nu^{\smile} \sqcup \nu;\kappa;\iota^{\smile};\lambda \sqcup \nu;\kappa;\kappa^{\smile};\nu^{\smile}\\ &= \lambda^{\smile};\lambda \sqcup \lambda^{\smile};\lambda;\nu;\nu^{\smile} \sqcup \nu;\nu^{\smile};\lambda^{\smile};\lambda \sqcup \nu;\nu^{\smile}\\ &= c_3 \sqcup c_3;\Theta \sqcup \Theta;c_3 \sqcup \Theta\\ &= c_3 \sqcup \Theta && \text{Lemma B.2.3}\\ \Xi;\Xi^{\smile} &= \Phi;\mathrm{X};\Psi^{\smile};\Psi;\mathrm{X}^{\smile};\Phi^{\smile}\\ &= \Phi;\mathrm{X};(c_3 \sqcup \Theta);\mathrm{X}^{\smile};\Phi^{\smile}\\ &= u_0;\Phi;\mathrm{X};c_3;\mathrm{X}^{\smile};\Phi^{\smile};u_0 \sqcup v_0;\Phi;\mathrm{X};\Theta;\mathrm{X}^{\smile};\Phi^{\smile};v_0\\ &= u_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};u_0 \sqcup v_0;\Phi;\Phi^{\smile};v_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};v_0 && \text{B.2.1, intf. pres., (iv)}\\ &= u_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};u_0 \sqcup v_0;\Phi;\Phi^{\smile};v_0;\Phi;\Phi^{\smile};v_0 && \text{(ii)}\\ &= u_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};u_0 \sqcup v_0;\mathsf{dom}\,\Phi && \text{Def. 6.2.1.iv)}\end{aligned}$$

Univalence of $u_0;\Phi$ shows $\Phi^{\smile};u_0;\Xi;\Xi^{\smile};u_0;\Phi = \Phi^{\smile};u_0;\Phi;\mathrm{X};\mathrm{X}^{\smile};\Phi^{\smile};u_0;\Phi = u_1;\mathrm{X};\mathrm{X}^{\smile};u_1$. The last term is idempotent because of (i), so we have

$$\begin{aligned}\mathrm{X};\mathrm{X}^{\smile} &= c_1 \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1 \sqcup v_1;\mathsf{dom}\,\mathrm{X}\\ &= c_1 \sqcup \Phi^{\smile};u_0;\Xi;\Xi^{\smile};u_0;\Phi \sqcup \mathsf{dom}\,(v_1;\mathrm{X})\\ &= c_1 \sqcup (\Phi^{\smile};u_0;\Xi;\Xi^{\smile};u_0;\Phi)^{+} \sqcup \mathsf{dom}\,(\Phi^{\smile};v_0;\Xi)\\ &= c_1;(\Phi^{\smile};u_0;\Xi)^{\triangleright};c_1 \sqcup \mathsf{dom}\,(\Phi^{\smile};v_0;\Xi)\end{aligned}$$



Here, we have used the following result, obtained using Lemma B.2.2:

$$\begin{aligned}\mathsf{dom}\,(\Phi^{\smile};v_0;\Xi) &= \mathsf{dom}\,(\Phi^{\smile};v_0;\Phi;\mathrm{X};\Psi^{\smile}) \;=\; \mathsf{dom}\,(\Phi^{\smile};v_0;\Phi;\mathrm{X})\\ &= \mathsf{dom}\,(v_1;\mathrm{X};\Theta) \;=\; \mathsf{dom}\,(v_1;\mathrm{X};\mathsf{dom}\,\Theta) \;=\; \mathsf{dom}\,(v_1;\mathrm{X};v_3) \;=\; \mathsf{dom}\,(v_1;\mathrm{X})\end{aligned}$$

$\Psi$ is total and injective:

$$\begin{aligned}\Psi;\Psi^{\smile} &= (\iota^{\smile};\lambda \sqcup \kappa^{\smile};\nu^{\smile});(\lambda^{\smile};\iota \sqcup \nu;\kappa)\\ &= \iota^{\smile};\lambda;\lambda^{\smile};\iota \sqcup \iota^{\smile};\lambda;\nu;\kappa \sqcup \kappa^{\smile};\nu^{\smile};\lambda^{\smile};\iota \sqcup \kappa^{\smile};\nu^{\smile};\nu;\kappa\\ &= \iota^{\smile};\iota \sqcup \iota^{\smile};\iota;\kappa^{\smile};\kappa \sqcup \kappa^{\smile};\kappa;\iota^{\smile};\iota \sqcup \kappa^{\smile};\kappa\\ &= \mathbb{I} && \text{univalence of } \iota \text{ and } \kappa\end{aligned}$$

$$\begin{aligned}\Phi^{\smile};u_0;\Xi &= \Phi^{\smile};u_0;\Phi;\mathrm{X};\Psi^{\smile} = u_1;\mathrm{X};\Psi^{\smile} && u_0;\Phi \text{ univalent}\\ \Xi^{\smile};u_0;\Phi;\Phi^{\smile};u_0;\Xi &= \Psi;\mathrm{X}^{\smile};u_1;\mathrm{X};\Psi^{\smile} \sqsubseteq \Psi;\Psi^{\smile} = \mathbb{I} && u_1;\mathrm{X} \text{ univalent}\end{aligned}$$

$$\Psi;\Psi^{\smile} = \mathbb{I} = c_2 \sqcup \mathsf{ran}\,(\Phi^{\smile};v_0;\Xi) = c_2;(\Phi^{\smile};u_0;\Xi)^{\triangleleft};c_2 \sqcup \mathsf{ran}\,(\Phi^{\smile};v_0;\Xi)$$

For this, we have used the following result:

$$\begin{aligned}\mathsf{ran}\,(\Phi^{\smile};v_0;\Xi) &= \mathsf{ran}\,(\Phi^{\smile};v_0;\Phi;\mathrm{X};\Psi^{\smile}) \;=\; \mathsf{ran}\,(v_1;\mathrm{X};\Theta;\Psi^{\smile})\\ &= \mathsf{ran}\,(v_1;\mathrm{X};\Theta;\nu;\kappa) \sqcup \mathsf{ran}\,(v_1;\mathrm{X};\Theta;\lambda^{\smile};\iota)\\ &= \mathsf{ran}\,(v_1;\mathrm{X};\nu;\kappa) \sqcup \mathsf{ran}\,(v_1;\mathrm{X};u_3;\lambda^{\smile};\iota) \;=\; \mathsf{ran}\,\kappa \sqcup \mathsf{ran}\,(v_3;u_3;\lambda^{\smile};\iota)\end{aligned}$$

For "alternative commutativity", we calculate:

$$\begin{aligned}\mathrm{X};\Psi^{\smile} &= \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} && \text{Lemma B.2.1}\\ &= u_1;\mathrm{X};\Psi^{\smile} \sqcup v_1;\mathrm{X};\Psi^{\smile}\\ &= u_1;\mathrm{X};\Psi^{\smile} \sqcup v_1;\mathrm{X};\Psi^{\smile};\Psi;\Psi^{\smile} && \Psi \text{ injective}\\ &= u_1;\mathrm{X};\Psi^{\smile} \sqcup v_1;\mathrm{X};\Theta;\Psi^{\smile}\\ &= \Phi^{\smile};u_0;\Xi \sqcup \Phi^{\smile};v_0;\Phi;\mathrm{X};\Psi^{\smile}\\ &= (\Phi^{\smile};u_0;\Xi)^{\boxast} \sqcup \Phi^{\smile};v_0;\Xi && \text{Def. } \Xi\end{aligned}$$

The combination property needs only slightly more attention than in the proof of Theorem 6.3.1:

$$\begin{aligned}\mathbb{I} &= \mathsf{ran}\,\mathrm{X} \sqcup \mathsf{ran}\,\Psi && \text{(ii), (iii) imply gluing cond.}\\ &= \mathsf{ran}\,(c_1;\mathrm{X}) \sqcup \mathsf{ran}\,(v_1;\mathrm{X}) \sqcup \mathsf{ran}\,(c_2;\Psi) \sqcup \mathsf{ran}\,(v_2;\Psi)\\ &= \mathsf{ran}\,(c_1;\mathrm{X}) \sqcup \mathsf{ran}\,(c_2;\Psi) \sqcup (\mathsf{ran}\,(v_1;\mathrm{X}) \sqcap \mathsf{ran}\,(v_2;\Psi)) && \mathsf{ran}\,(v_1;\mathrm{X}) = v_3 = \mathsf{ran}\,(v_2;\Psi)\\ &= \mathsf{ran}\,(c_1;\mathrm{X}) \sqcup \mathsf{ran}\,(c_2;\Psi) \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \mathsf{ran}\,(v_2;\Psi)) && \mathsf{ran}\,(v_2;\Psi) \sqsubseteq \mathsf{ran}\,(v_1;\mathrm{X})\\ &= \mathsf{ran}\,(c_1;\mathrm{X}) \sqcup \mathsf{ran}\,(c_2;\Psi) \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Theta) && \text{(iv)}\\ &= \mathrm{X}^{\smile};c_1;\mathrm{X} \sqcup \Psi^{\smile};c_2;\Psi \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) && c_1;\mathrm{X} \text{ univalent, } \Psi^{\smile};\Psi = c_3 \sqcup \Theta\end{aligned}$$

Standard commutativity uses (iii) and interface preservation of $\Phi$ in the last step:

$$\begin{aligned}\Xi;\Psi &= \Phi;\mathrm{X};\Psi^{\smile};\Psi \;=\; \Phi;\mathrm{X};c_3 \sqcup \Phi;\mathrm{X};\Theta\\ &= \Phi;\mathrm{X};c_3 \sqcup \Phi;\Phi^{\smile};v_0;\Phi;\mathrm{X} \;=\; \Phi;\mathrm{X};c_3 \sqcup v_0;\Phi;\mathrm{X};v_2 \;=\; \Phi;\mathrm{X}\end{aligned}$$

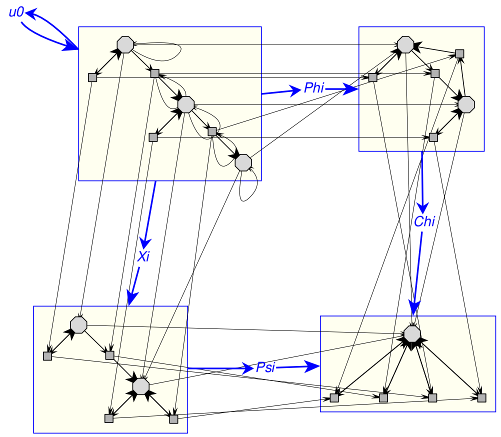
The proof that with the additional conditions $\Xi$ is a standard gluing morphism proceeds in essentially the same way as in the proof of Theorem 6.3.1. □

## B.3 Monomorphy of Weak Pullouts

The ease with which monomorphy of pullouts (Theorem 6.2.4) could be proved essentially depended on the fact that both X and Ψ were almost-injective on the respective parameter parts.

In the case of weak pullouts, we do not have almost-injectivity of X on the whole of the parameter part, and therefore the argument will be more complicated.

In this context it is important to note that, given only almost-injectivity of Φ on $v_0$ besides $u_0$, neither the pullout nor the weak pullout guarantee "strong alternative parameter commutativity" $v_1;\mathrm{X};\Psi^{\smile};v_2 = \Phi^{\smile};v_0;\Xi$, because identifications between border nodes of the parameter images may be induced by other parts of the interface. Consider the following situation, where the interface $u_0$ consists of everything but the two loop edges, and the interface node that is not incident with one of the variables is identified with one border node via Φ, and with the other border node via Ξ:




This forces the result of the (weak) pullout construction to identify the two border nodes, which implies that the inclusion $\Phi^{\smile};v_0;\Xi \sqsubseteq v_1;\mathrm{X};\Psi^{\smile};v_2$ is strict in this case.

Outside the borders, however, we do have parameter commutativity via the parameter part of the gluing object.

**Lemma B.3.1** In weak pullouts, we have *alternative parameter commutativity*:

$$v_1;\mathrm{X};\Psi^{\smile};v_2 = \Phi^{\smile};v_0;\Xi \sqcup b_1;\mathrm{X};\Psi^{\smile};b_2 \ .$$

**Proof:**

$$\begin{aligned}
& v_1;\mathrm{X};\Psi^{\smile};v_2 \\
= \; & v_1;(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};\Phi^{\smile};\Xi;v_2 \\
= \; & v_1;\Phi^{\smile};\Xi;v_2 \sqcup v_1;(\Phi^{\smile};u_0;f_0;\Xi;\Xi^{\smile};u_0;f_0;\Phi)^{+};\Phi^{\smile};\Xi;v_2 \\
= \; & v_1;\Phi^{\smile};v_0;\Xi;v_2 \sqcup v_1;\Phi^{\smile};u_0;\Xi;v_2 \sqcup v_1;u_1;(\Phi^{\smile};u_0;f_0;\Xi;\Xi^{\smile};u_0;f_0;\Phi)^{+};\Phi^{\smile};\Xi;u_2;v_2 \\
= \; & v_1;\Phi^{\smile};v_0;\Xi;v_2 \sqcup v_1;u_1;\Phi^{\smile};u_0;\Xi;u_2;v_2 \sqcup b_1;(\Phi^{\smile};u_0;f_0;\Xi;\Xi^{\smile};u_0;f_0;\Phi)^{+};\Phi^{\smile};\Xi;b_2 \\
= \; & \Phi^{\smile};v_0;\Xi \sqcup b_1;\Phi^{\smile};\Xi;b_2 \sqcup b_1;(\Phi^{\smile};u_0;f_0;\Xi;\Xi^{\smile};u_0;f_0;\Phi)^{+};\Phi^{\smile};\Xi;b_2 \\
= \; & \Phi^{\smile};v_0;\Xi \sqcup b_1;\mathrm{X};\Psi^{\smile};b_2
\end{aligned}$$

□



**Lemma B.3.2** [←189]

$$c_1;\mathrm{X};\Psi^{\smile} = c_1;\mathrm{X};\Psi^{\smile};u_2 \ , \qquad c_1;\mathrm{X};\mathrm{X}^{\smile} = c_1;\mathrm{X};\mathrm{X}^{\smile};u_1 \ , \qquad c_2;\Psi;\Psi^{\smile} = c_2;\Psi;\Psi^{\smile};u_2 \ .$$

**Proof**: Obvious from alternative commutativity and the semi-injectivity properties together with Def. 6.1.2.ii). □

The following lemma is the only place that employs the possible presence of sharp products. It is open whether a proof without product exists; however, we conjecture that the typical "cut-across diamond shape" of the constellation in the proof makes this impossible.

**Lemma B.3.3** [←147, 188, 189] If the Dedekind category underlying the discussion may be embedded in a Dedekind category with sharp products, the following holds:

$$\mathrm{X}^{\smile};v_1;\mathrm{X};\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi \sqsubseteq \mathbb{I}$$

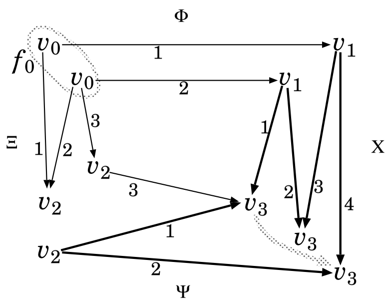
**Proof**: First we have:

$$\begin{aligned}
& \mathrm{X}^{\smile};v_1;\mathrm{X};\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi \\
\sqsubseteq \; & \mathrm{X}^{\smile};v_1;(\mathbb{I} \sqcup \Phi^{\smile};v_0;\Xi;f_2;\Xi^{\smile};v_0;\Phi \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1);v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi \\
= \; & (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \sqcup (\mathrm{X}^{\smile};v_1;\Phi^{\smile};v_0;\Xi;f_2;\Xi^{\smile};v_0;\Phi;v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \sqcup \\
& (\mathrm{X}^{\smile};v_1;u_1;\mathrm{X};\mathrm{X}^{\smile};u_1;v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \\
= \; & v_3 \sqcup (\mathrm{X}^{\smile};v_1;\Phi^{\smile};v_0;\Xi;f_2;\Xi^{\smile};v_0;\Phi;v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \sqcup (\mathrm{X}^{\smile};b_1;\mathrm{X};\mathrm{X}^{\smile};b_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \\
\sqsubseteq \; & v_3 \sqcup (\Psi^{\smile};\Xi^{\smile};v_0;\Xi;f_2;\Xi^{\smile};v_0;\Phi;v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi) \sqcup (\mathbb{I};\mathbb{I} \sqcap \Psi^{\smile};v_2;\Psi) \\
= \; & \{ \ \Psi \text{ almost-inj, besides } u_2\text{, and } u_2\text{-component covered by } b_1\text{-term} \ \} \\
& \mathbb{I} \sqcup \Psi^{\smile};(\Xi^{\smile};v_0;\Xi;f_2;\Xi^{\smile};v_0;\Phi;v_1;\mathrm{X} \sqcap v_2;\Psi)
\end{aligned}$$

This calculation mainly served to show that everything where a non-empty intersection with $u_0$, $u_1$, or $u_2$ is involved, is already contained in the identity. Therefore, we may restrict further considerations to the $v$-parts only — even in cases like using almost-injectivity of Ψ besides $u_2$, the $u$-component may be ignored since the $u$ would always propagate to the outside and there it is already covered.

We now give a sketch of a graphical proof — this kind of proof is justified by having a translation into the language of allegories with sharp products (see e.g. [Kah96]), so this is the reason we need sharp products in the assumption.

The proof is explained along the following drawing:



When writing e.g. $\Phi_1$, this indicates that we refer to the arrow numbered 1 in the $\Phi$-group. Otherwise, the relational expressions are to be read as if these indices were absent.

- Start with a graph (with bold edges) representing the expression

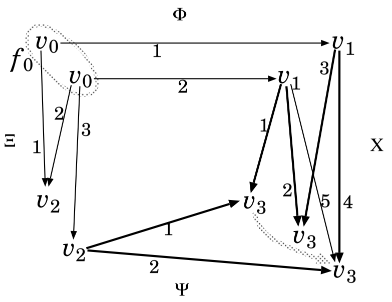
$$X_1^\smile{;}v_1{;}X_2{;}X_3^\smile{;}v_1{;}X_4 \sqcap \Psi_1^\smile{;}v_2{;}\Psi_2 \ .$$

  The thick grey arrow represents this expression as a whole.

- Via semi-injectivity of X (ignoring the $u$-parts) we get: $X_2{;}X_3^\smile = \Phi_2^\smile{;}f_0{;}\Xi_2{;}\Xi_1^\smile{;}f_0{;}\Phi_1$ .

- Commutativity: $\Phi_2{;}X_1 = \Xi_3{;}\Psi_3$

- Now, almost-injectivity of $\Psi$ gives us $\Psi_3{;}\Psi_1^\smile = \mathbb{I}$ (up to $u_2$, which may be ignored, as explained above). This allows us to identify the arrows $\Psi_3$ and $\Psi_1$, and their source vertices:




- Univalence of $\Phi$ on $f_0$ together with commutativity now produces. $\Xi_3{;}\Psi_2 = \Phi_2{;}X_5$

- Finally, parameter tabulation gives us $X_1^\smile{;}v_1{;}X_5 \sqcap \Psi_1^\smile{;}v_2{;}\Psi_2 \sqsubseteq \mathbb{I}$. This identifies source and target of the thick grey arrow and thus completes the proof of the lemma. □

**Proof of Theorem 6.5.6**: Assume that for the reasonable gluing setup $(G_0, u_0, v_0, \Xi, \Phi)$, there are two weak pullouts $G_2 \xrightarrow{\Psi} G_3 \xleftarrow{X} G_1$ and $G_2 \xrightarrow{\Psi'} G_3 \xleftarrow{X'} G_1$. Then we define:

$$Y := X^\smile{;}c_1{;}X' \sqcup \Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim{;}\Psi' \sqcup (X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi')$$

Note that the weak pullout properties imply the following:

$$\begin{array}{rclcrclcrcl} X{;}\Psi^\smile &=& X'{;}\Psi'^\smile & & X{;}X^\smile &=& X'{;}X'^\smile & & \mathsf{dom}\,X &=& \mathsf{dom}\,X' \\ & & & & \Psi{;}\Psi^\smile &=& \Psi'{;}\Psi'^\smile & & \mathsf{dom}\,\Psi &=& \mathsf{dom}\,\Psi' \end{array}$$

When using these properties, we refer at most to the corresponding weak pullout property.



Totality:

$$\begin{array}{cl}
& \mathsf{dom}\,Y \\
= & \mathsf{dom}\,(X^\smile{;}c_1{;}X') \sqcup \mathsf{dom}\,(\Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim{;}\Psi') \sqcup \mathsf{dom}\,(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi') \\
= & \mathsf{dom}\,(X^\smile{;}c_1{;}\mathsf{dom}\,X') \sqcup \mathsf{dom}\,(\Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim{;}\mathsf{dom}\,\Psi') \sqcup \mathsf{dom}\,(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi') \\
= & \quad \{ \text{ semi-injectivity } \} \\
& \mathsf{dom}\,(X^\smile{;}c_1{;}\mathsf{dom}\,X) \sqcup \mathsf{dom}\,(\Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim{;}\mathsf{dom}\,\Psi) \sqcup \mathsf{dom}\,(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi') \\
= & \mathsf{dom}\,(X^\smile{;}c_1) \sqcup \mathsf{dom}\,(\Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim) \sqcup \mathsf{dom}\,(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi') \\
= & \mathsf{ran}\,(c_1{;}X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^\sim{;}\Psi) \sqcup \mathsf{dom}\,(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi') \\
= & \quad \{ \text{ Lemma A.2.2.iv) } \} \\
& \mathsf{ran}\,(c_1{;}X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^\sim{;}\Psi) \sqcup \mathsf{dom}\,(X^\smile{;}v_1{;}X'{;}\Psi'^\smile{;}v_2 \sqcap \Psi^\smile) \\
\sqsupseteq & \quad \{ \text{ alternative parameter commutativity } \} \\
& \mathsf{ran}\,(c_1{;}X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^\sim{;}\Psi) \sqcup \mathsf{dom}\,(X^\smile{;}\Phi^\smile{;}v_0{;}\Xi \sqcap \Psi^\smile) \\
= & \quad \{ \text{ Lemma A.2.2.iv) } \} \\
& \mathsf{ran}\,(c_1{;}X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^\sim{;}\Psi) \sqcup \mathsf{dom}\,(X^\smile{;}\Phi^\smile{;}v_0 \sqcap \Psi^\smile{;}\Xi^\smile) \\
= & \quad \{ \text{ commutativity } \} \\
& \mathsf{ran}\,(c_1{;}X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^\sim{;}\Psi) \sqcup \mathsf{dom}\,(\Psi^\smile{;}\Xi^\smile{;}v_0 \sqcap \Psi^\smile{;}\Xi^\smile) \\
= & \mathsf{ran}\,(c_1{;}X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^\sim{;}\Psi) \sqcup \mathsf{ran}\,(v_2{;}\Psi) \\
= & \quad \{ \text{ combination } \} \\
& \mathbb{I}
\end{array}$$

Factorisation: We start with factorisation of $\Psi'$.

$$\Psi{;}Y = \Psi{;}X^\smile{;}c_1{;}X' \sqcup \Psi{;}\Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim{;}\Psi' \sqcup \Psi{;}(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi')$$

For the first term we have:

$$\Psi{;}X^\smile{;}c_1{;}X' = \Psi'{;}X'^\smile{;}c_1{;}X' \sqsubseteq \Psi'$$

For the second term, $(\mathsf{ran}\,\Xi)^\sim \sqsubseteq q_2 \sqsubseteq \mathsf{dom}\,\Psi$ and univalence of $\Psi'$ on $(\mathsf{ran}\,\Xi)^\sim$ give us:

$$(\mathsf{ran}\,\Xi)^\sim{;}\Psi' \sqsubseteq \Psi{;}\Psi^\smile{;}(\mathsf{ran}\,\Xi)^\sim{;}\Psi' \sqsubseteq \Psi'$$

For the third term, we have:

$$\begin{array}{rcll}
v_2{;}\Psi' &=& v_2{;}\Psi'{;}X'^\smile{;}v_1{;}X' \sqcap v_2{;}\Psi' & \text{parameter tabulation and commutativity} \\
&=& v_2{;}\Psi{;}X^\smile{;}v_1{;}X' \sqcap v_2{;}\Psi' & \text{alternative commutativity} \\
&\sqsubseteq& \Psi{;}(X^\smile{;}v_1{;}X' \sqcap \Psi^\smile{;}v_2{;}\Psi') & \text{modal rule} \\
&\sqsubseteq& \Psi{;}\Psi^\smile{;}v_2{;}\Psi' & \\
&\sqsubseteq& v_2{;}\Psi' \sqcup u_2{;}\Psi{;}\Psi^\smile{;}u_2{;}v_2{;}\Psi' & \Psi \text{ almost injective besides } u_2 \\
&=& v_2{;}\Psi' \sqcup u_2{;}\Psi'{;}\Psi'^\smile{;}b_2{;}\Psi' & \text{semi-inj. } \Psi\text{, Def. } b_2 \\
&\sqsubseteq& v_2{;}\Psi' \sqcup u_2{;}\Psi' & \Psi' \text{ univalent on } b_2 \\
&\sqsubseteq& \Psi' &
\end{array}$$

So altogether we have $\Psi{;}Y \sqsubseteq \Psi'$ and:

$$\begin{array}{cll}
& \Psi{;}Y & \\
\sqsupseteq & \Psi'{;}\mathsf{ran}\,(c_1{;}X') \sqcup (\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' \sqcup v_2{;}\Psi' & \\
= & \Psi'{;}\mathsf{ran}\,(c_1{;}X') \sqcup w_2{;}\Psi' & \text{Def. } w_2 \\
= & \Psi'{;}\mathsf{ran}\,(c_1{;}X') \sqcup w_2{;}\Psi'{;}\mathsf{ran}\,(w_2{;}\Psi') & \\
= & u_2{;}\Psi'{;}\mathsf{ran}\,(c_1{;}X') \sqcup w_2{;}\Psi'{;}\mathsf{ran}\,(c_1{;}X') \sqcup w_2{;}\Psi'{;}\mathsf{ran}\,(w_2{;}\Psi') & \\
= & u_2{;}\Psi' \sqcup w_2{;}\Psi'{;}(\mathsf{ran}\,(c_1{;}X') \sqcup \mathsf{ran}\,(w_2{;}\Psi')) & \mathsf{ran}\,(u_2{;}\Psi') \sqsubseteq \mathsf{ran}\,(c_1{;}X') \\
= & u_2{;}\Psi' \sqcup w_2{;}\Psi' & \text{result coverage} \\
= & \Psi' \;, &
\end{array}$$

which shows $\Psi{;}Y = \Psi'$. For factorisation of $X'$ we proceed in the same way:

$$X{;}Y = X{;}X^{\smile}{;}c_1{;}X' \sqcup X{;}\Psi^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' \sqcup X{;}(X^{\smile}{;}v_1{;}X' \sqcap \Psi^{\smile}{;}v_2{;}\Psi')$$

For the first part we use semi-injectivity of X and univalence of $X'$ on $c_1$:

$$X{;}X^{\smile}{;}c_1{;}X' = X'{;}X'^{\smile}{;}c_1{;}X' = X'{;}\mathsf{ran}\,(c_1{;}X')$$

For the second part, alternative commutativity and univalence of $\Psi'$ on $(\mathsf{ran}\,\Xi)^{\smile}$ immediately yield:

$$X{;}\Psi^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' = X'{;}\Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' \sqsubseteq X'$$

For the third part we have:

$$\begin{array}{cll}
& v_1{;}X' & \\
\sqsubseteq & X{;}X^{\smile}{;}v_1{;}X' & \mathsf{dom}\,X = \mathsf{dom}\,X' \\
\sqsubseteq & X{;}(X^{\smile}{;}v_1{;}X' \sqcap \Psi^{\smile}{;}v_2{;}\Psi') & \\
\sqsubseteq & X{;}X^{\smile}{;}v_1{;}X' \sqcap X{;}\Psi^{\smile}{;}v_2{;}\Psi' & \\
= & X{;}X^{\smile}{;}v_1{;}X' \sqcap (c_1 \sqcup v_1){;}X{;}\Psi^{\smile}{;}v_2{;}\Psi' & \\
\sqsubseteq & c_1{;}X{;}\Psi^{\smile}{;}v_2{;}\Psi' \sqcup (X{;}X^{\smile}{;}v_1{;}X' \sqcap v_1{;}X{;}\Psi^{\smile}{;}v_2{;}\Psi') & \\
= & c_1{;}X'{;}\Psi'^{\smile}{;}u_2{;}v_2{;}\Psi' \sqcup (X{;}X^{\smile}{;}v_1{;}X' \sqcap v_1{;}X{;}\Psi^{\smile}{;}v_2{;}\Psi') & \text{Def. 6.1.2.ii), alt. intf. comm.} \\
\sqsubseteq & c_1{;}X' \sqcup (X{;}X^{\smile}{;}v_1{;}X' \sqcap v_1{;}X{;}\Psi^{\smile}{;}v_2{;}\Psi') & b_2{;}\Psi' \text{ unival.} \\
= & c_1{;}X' \sqcup (X{;}X^{\smile}{;}v_1{;}X' \sqcap v_1{;}X'{;}\Psi'^{\smile}{;}v_2{;}\Psi') & \text{alt. comm.} \\
\sqsubseteq & c_1{;}X' \sqcup v_1{;}X'{;}(X'^{\smile}{;}v_1{;}X{;}X^{\smile}{;}v_1{;}X' \sqcap \Psi'^{\smile}{;}v_2{;}\Psi') & \text{modal rule} \\
\sqsubseteq & c_1{;}X' \sqcup v_1{;}X' & \text{Lemma B.3.3} \\
= & X' &
\end{array}$$

Altogether, with an argument essentially as above, we therefore have $X{;}Y = X'$.
*Univalence of* Y*:*

$$\begin{array}{rcl}
Y^{\smile}{;}Y & = & (X'^{\smile}{;}c_1{;}X \sqcup \Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi \sqcup (X'^{\smile}{;}v_1{;}X \sqcap \Psi'^{\smile}{;}v_2{;}\Psi)){;} \\
& & (X^{\smile}{;}c_1{;}X' \sqcup \Psi^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' \sqcup (X^{\smile}{;}v_1{;}X' \sqcap \Psi^{\smile}{;}v_2{;}\Psi'))
\end{array}$$



Since we want to show inclusion in identity, we need only consider six of the nine terms resulting from this composition; the three omitted terms are the converses of the three considered mixed terms.

$$\begin{array}{cll}
& X'^{\smile}{;}c_1{;}X{;}X^{\smile}{;}c_1{;}X' & \\
= & X'^{\smile}{;}c_1{;}X'{;}X'^{\smile}{;}c_1{;}X' & \text{semi-injectivity of X} \\
\sqsubseteq & \mathbb{I} & X' \text{ univalent on } c_1 \\
& X'^{\smile}{;}c_1{;}X{;}\Psi^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' & \\
= & X'^{\smile}{;}c_1{;}X'{;}\Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' & \text{alternative commutativity} \\
\sqsubseteq & \mathbb{I} & X' \text{ univalent on } c_1 \text{ and } \Psi' \text{ on } (\mathsf{ran}\,\Xi)^{\smile} \\
& X'^{\smile}{;}c_1{;}X{;}(X^{\smile}{;}v_1{;}X' \sqcap \Psi^{\smile}{;}v_2{;}\Psi') & \\
\sqsubseteq & X'^{\smile}{;}c_1{;}X{;}X^{\smile}{;}v_1{;}X' & \\
= & X'^{\smile}{;}c_1{;}X'{;}X'^{\smile}{;}c_1{;}v_1{;}X' & \text{Lemma B.3.2, semi-injectivity of X} \\
\sqsubseteq & \mathbb{I} & X' \text{ univalent on } c_1 \\
& \Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi{;}\Psi^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' & \\
= & \Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi'{;}\Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi' & \text{semi-injectivity of } \Psi \\
\sqsubseteq & \mathbb{I} & \Psi' \text{ univalent on } (\mathsf{ran}\,\Xi)^{\smile} \\
& \Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi{;}(X^{\smile}{;}v_1{;}X' \sqcap \Psi^{\smile}{;}v_2{;}\Psi') & \\
\sqsubseteq & \Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi{;}\Psi^{\smile}{;}v_2{;}\Psi' & \\
= & \Psi'^{\smile}{;}(\mathsf{ran}\,\Xi)^{\smile}{;}\Psi'{;}\Psi'^{\smile}{;}c_2{;}v_2{;}\Psi' & \text{Lemma B.3.2, semi-injectivity of } \Psi \\
\sqsubseteq & \Psi'^{\smile}{;}((\mathsf{ran}\,\Xi)^{\smile} \sqcup b_2){;}\Psi' & \Psi' \text{ univalent on } (\mathsf{ran}\,\Xi)^{\smile} \\
\sqsubseteq & \mathbb{I} & \Psi' \text{ univalent on } (\mathsf{ran}\,\Xi)^{\smile} \sqcup b_2
\end{array}$$

$$\begin{array}{cll}
& (X'^{\smile}{;}v_1{;}X \sqcap \Psi'^{\smile}{;}v_2{;}\Psi){;}(X^{\smile}{;}v_1{;}X' \sqcap \Psi^{\smile}{;}v_2{;}\Psi') & \\
\sqsubseteq & X'^{\smile}{;}v_1{;}X{;}X^{\smile}{;}v_1{;}X' \sqcap \Psi'^{\smile}{;}v_2{;}\Psi{;}\Psi^{\smile}{;}v_2{;}\Psi' & \\
= & X'^{\smile}{;}v_1{;}X{;}X^{\smile}{;}v_1{;}X' \sqcap (\Psi'^{\smile}{;}v_2\Psi' \sqcup \Psi'^{\smile}{;}v_2{;}u_2{;}\Psi{;}\Psi^{\smile}{;}u_2{;}v_2{;}\Psi') & \Psi \text{ almost-inj. besides } u_2 \\
= & X'^{\smile}{;}v_1{;}X{;}X^{\smile}{;}v_1{;}X' \sqcap (\Psi'^{\smile}{;}v_2\Psi' \sqcup \Psi'^{\smile}{;}b_2{;}\Psi'{;}\Psi'^{\smile}{;}b_2{;}\Psi') & \text{semi-injectivity} \\
\sqsubseteq & (X'^{\smile}{;}v_1{;}X{;}X^{\smile}{;}v_1{;}X' \sqcap \Psi'^{\smile}{;}v_2\Psi') \sqcup \mathbb{I} & \Psi' \text{ univalent on } b_2 \\
\sqsubseteq & \mathbb{I} & \text{Lemma B.3.3}
\end{array}$$

Up to now, we have shown that Y is a mapping that factorises $X'$ and $\Psi'$. Since the same argument is also valid for $Y^{\smile}$, it is already shown that Y is bijective, too.

*Uniqueness:* Assume a total relation $\mathrm{Y}' : G_3 \to G_4$ with $\Psi;\mathrm{Y}' = \Psi'$ and $\mathrm{X};\mathrm{Y}' = \mathrm{X}'$. Then

$$\begin{array}{lll}
& \mathrm{Y}' & \\
= & (\mathsf{ran}\,(c_1;\mathrm{X}) \sqcup \mathsf{ran}\,(w_2;\Psi));\mathrm{Y}' & \text{result coverage} \\
= & \mathrm{X}^{\smile};c_1;\mathrm{X};\mathrm{Y}' \sqcup \mathsf{ran}\,(w_2;\Psi);\mathrm{Y}' & c_1;\mathrm{X} \text{ univalent} \\
= & \mathrm{X}^{\smile};c_1;\mathrm{X};\mathrm{Y}' \sqcup \mathsf{ran}\,(((\mathsf{ran}\,\Xi)^{\sim});\Psi);\mathrm{Y}' \sqcup \mathsf{ran}\,(v_2;\Psi);\mathrm{Y}' & \\
= & \mathrm{X}^{\smile};c_1;\mathrm{X};\mathrm{Y}' \sqcup \Psi^{\smile};((\mathsf{ran}\,\Xi)^{\sim});\Psi;\mathrm{Y}' \sqcup (\mathbb{I} \sqcap \Psi^{\smile};v_2;\Psi);\mathrm{Y}' & ((\mathsf{ran}\,\Xi)^{\sim});\Psi \text{ unival.} \\
= & \mathrm{X}^{\smile};c_1;\mathrm{X};\mathrm{Y}' \sqcup \Psi^{\smile};((\mathsf{ran}\,\Xi)^{\sim});\Psi;\mathrm{Y}' \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X} \sqcap \Psi^{\smile};v_2;\Psi);\mathrm{Y}' & \mathsf{ran}\,(v_2;\Psi) = v_3 \\
\sqsubseteq & \mathrm{X}^{\smile};c_1;\mathrm{X};\mathrm{Y}' \sqcup \Psi^{\smile};((\mathsf{ran}\,\Xi)^{\sim});\Psi;\mathrm{Y}' \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X};\mathrm{Y}' \sqcap \Psi^{\smile};v_2;\Psi;\mathrm{Y}') & \\
= & \mathrm{X}^{\smile};c_1;\mathrm{X}' \sqcup \Psi^{\smile};((\mathsf{ran}\,\Xi)^{\sim});\Psi' \sqcup (\mathrm{X}^{\smile};v_1;\mathrm{X}' \sqcap \Psi^{\smile};v_2;\Psi') & \text{factorisation via } \mathrm{Y}' \\
= & \mathrm{Y} &
\end{array}$$

Since Y is univalent, totality of $\mathrm{Y}'$ gives us equality. □

## B.4 Correctness of the Direct Result Construction

This section is dedicated to the proof that the direct result construction of Def. 6.5.7 produces a weak pullout as defined in Def. 6.5.5.

Def. 6.1.2.ii) allow us to provide equivalent, but longer shapes for $Q_3$ and $Q_4$; these are sometimes useful for technical reasons:

**Lemma B.4.1**

$$\begin{array}{lllll}
Q_3 & = & \iota^{\smile};\Phi^{\smile};\Xi;\lambda & = & \iota^{\smile};\Phi^{\smile};u_0;\Xi;\lambda \\
Q_4 & = & \iota^{\smile};\Phi^{\smile};y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};\Phi;\iota & = & \iota^{\smile};\Phi^{\smile};u_0;y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota
\end{array}$$ □

First we show commutativity:

**Lemma B.4.2** $\Phi;\mathrm{X} = \Xi;\Psi$

**Proof**: We show this equality as conjunction of two inclusions.

$$\begin{array}{llll}
\Phi;\mathrm{X} \sqsubseteq \Xi;\Psi & \Leftrightarrow & \Phi;\mathrm{X}_0;\theta \sqsubseteq \Xi;\Psi_0;\theta & \\
& \Leftrightarrow & \Phi;\mathrm{X}_0 \sqsubseteq \Xi;\Psi_0;\theta;\theta^{\smile} & \text{Lemma A.1.2.iii)} \\
& \Leftrightarrow & \Phi;\mathrm{X}_0 \sqsubseteq \Xi;\Psi_0;\Theta & \\
& \Leftrightarrow & \Phi;(\iota \sqcup \Phi^{\smile};\Xi;\lambda \sqcup \pi^{\smile};\kappa) \sqsubseteq \Xi;\Psi_0;\Theta & \\
& \Leftrightarrow & \Phi;\iota \sqcup \Phi;\Phi^{\smile};\Xi;\lambda \sqcup \Phi;\pi^{\smile};\kappa \sqsubseteq \Xi;\Psi_0;\Theta & \\
& \Leftrightarrow & u_0;\Phi;\iota \sqcup v_0;\Phi;\iota \sqcup \Phi;\Phi^{\smile};\Xi;\lambda \sqcup \Phi;\pi^{\smile};\kappa \sqsubseteq \Xi;\Psi_0;\Theta &
\end{array}$$

Now we show this last inclusion by showing inclusions for every component of the join on the left-hand side. We shall frequently rely on $u_0 \sqsubseteq \mathsf{dom}\,\Xi$ and on the tabulation properties of $\pi$ and $\rho$, sometimes without mention.

$$\begin{array}{lll}
u_0;\Phi;\iota & \sqsubseteq & \Xi;\Xi^{\smile};u_0;\Phi;\iota \ \sqsubseteq\ \Xi;\Psi_0 \ \sqsubseteq\ \Xi;\Psi_0;\Theta \\
v_0;\Phi;\iota & = & v_0;u_0;\Phi;\iota \ \sqsubseteq\ v_0;\Xi;\Xi^{\smile};v_0;\Phi;\iota \ \sqsubseteq\ \Xi;v_2;\Xi^{\smile};v_0;\Phi;v_1;\iota \\
& = & \Xi;\rho^{\smile};\pi;\iota \ =\ \Xi;\rho^{\smile};\kappa;\kappa^{\smile};\pi;\iota \ \sqsubseteq\ \Xi;\Psi_0;Q_1
\end{array}$$



$$\begin{array}{llll}
\Phi;\Phi^{\smile};\Xi;\lambda & = & \Phi;\Phi^{\smile};u_0;\Xi;\lambda \sqcup \Phi;\Phi^{\smile};v_0;\Xi;\lambda & \\
& \sqsubseteq & \Phi;u_1;\Phi^{\smile};u_0;\Xi;\lambda \sqcup v_0;\Xi;\lambda \sqcup u_0;\Phi;\Phi^{\smile};u_0;v_0;\Xi;\lambda & \Phi \text{ almost-inj. bes. } u_0 \\
& = & u_0;\Phi;u_1;\Phi^{\smile};u_0;\Xi;\lambda \sqcup v_0;\Xi;\lambda & \text{Def. 6.1.2.ii)} \\
& \sqsubseteq & \Xi;\Xi^{\smile};u_0;\Phi;u_1;\Phi^{\smile};u_0;\Xi;\lambda \sqcup \Xi;\Psi_0 & u_0 \sqsubseteq \mathsf{dom}\,\Xi \\
& \sqsubseteq & \Xi;\Xi^{\smile};u_0;\Phi;\iota;\iota^{\smile};\Phi^{\smile};u_0;\Xi;\lambda \sqcup \Xi;\Psi_0 & \mathsf{ran}\,(u_0;\Phi) \sqsubseteq \mathsf{dom}\,\iota \\
& \sqsubseteq & \Xi;\Psi_0;Q_3 \sqcup \Xi;\Psi_0 & \text{Def. } \Psi_0\text{, Def. } Q_3 \\
& \sqsubseteq & \Xi;\Psi_0;\Theta &
\end{array}$$

For the last term we use $u_0 \sqcup v_0 = \mathbb{I}$ again:

$$\begin{array}{llll}
u_0;\Phi;\pi^{\smile};\kappa & \sqsubseteq & \Xi;\Xi^{\smile};u_0;\Phi;\pi^{\smile};\kappa & u_0 \sqsubseteq \mathsf{dom}\,\Xi \\
& = & \Xi;\Xi^{\smile};u_0;\Phi;\iota;\iota^{\smile};\pi^{\smile};\kappa & \mathsf{ran}\,(u_0;\Phi) \sqsubseteq \mathsf{dom}\,\iota \\
& \sqsubseteq & \Xi;\Psi_0;\iota^{\smile};\pi^{\smile};\kappa & \text{Def. } \Psi_0 \\
& \sqsubseteq & \Xi;\Psi_0;Q_1 & \text{Def. } Q_1 \\
& \sqsubseteq & \Xi;\Psi_0;\Theta & \\
v_0;\Phi;\pi^{\smile};\kappa & \sqsubseteq & v_0;\Phi;\pi^{\smile};\rho;\rho^{\smile};\kappa & \\
& = & v_0;\Phi;\Phi^{\smile};v_0;\Xi;\rho^{\smile};\kappa & \text{tabulation} \\
& \sqsubseteq & u_0;\Phi;\pi^{\smile};\kappa \sqcup v_0;\Xi;\rho^{\smile}\kappa & \Phi \text{ almost-injective besides } u_0 \\
& \sqsubseteq & \Xi;\Psi_0;\Theta \sqcup v_0;\Xi;\Psi_0 & \text{above, and Def. } \Psi_0 \\
& = & \Xi;\Psi_0;\Theta &
\end{array}$$

For the opposite inclusion, we have:

$$\begin{array}{lll}
& \Xi;\Psi \sqsubseteq \Phi;\mathrm{X} & \\
\Leftrightarrow & \Xi;\Psi_0;\theta \sqsubseteq \Phi;\mathrm{X}_0;\theta & \\
\Leftrightarrow & \Xi;\Psi_0 \sqsubseteq \Phi;\mathrm{X}_0;\theta;\theta^{\smile} & \text{Lemma A.1.2.iii)} \\
\Leftrightarrow & \Xi;\Psi_0 \sqsubseteq \Phi;\mathrm{X}_0;\Theta & \\
\Leftrightarrow & \Xi;(\Xi^{\smile};u_0;\Phi;\iota \sqcup \rho^{\smile};\kappa \sqcup \lambda) \sqsubseteq \Phi;\mathrm{X}_0;\Theta & \\
\Leftrightarrow & \Xi;\Xi^{\smile};u_0;\Phi;\iota \sqcup \Xi;\rho^{\smile};\kappa \sqcup \Xi;\lambda \sqsubseteq \Phi;\mathrm{X}_0;\Theta &
\end{array}$$

We show this as separate inclusions, again:

$$\begin{array}{lll}
& \Xi;\Xi^{\smile};u_0;\Phi;\iota & \\
= & (y_0 \sqcup u_0;y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim});u_0;\Phi;\iota & \text{image preservation of } \Xi\text{, Def. } y_0 \\
\sqsubseteq & \Phi;\iota \sqcup u_0;y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota & \\
\sqsubseteq & \Phi;\mathrm{X}_0 \sqcup u_0;y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota & \\
\sqsubseteq & \Phi;\mathrm{X}_0 \sqcup \Phi;c_1;\Phi^{\smile};u_0;y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota & u_0 \sqsubseteq \mathsf{dom}\,\Phi \text{ and } \mathsf{ran}\,(u_0;\Phi) \sqsubseteq c_1 \\
= & \Phi;\mathrm{X}_0 \sqcup \Phi;\iota;\iota^{\smile};\Phi^{\smile};u_0;y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota & \\
= & \Phi;\mathrm{X}_0 \sqcup \Phi;\iota;Q_4 & \text{Def. } Q_4 \\
\sqsubseteq & \Phi;\mathrm{X}_0;\Theta &
\end{array}$$

$$
\begin{array}{lll}
 & \Xi;\rho^{\smile};\kappa & \\
= & u_0;\Xi;\rho^{\smile};\kappa \sqcup v_0;\Xi;\rho^{\smile};\kappa & \\
= & u_0;\Xi;\rho^{\smile};\kappa \sqcup v_0;\Xi;f_2;\rho^{\smile};\kappa \sqcup v_0;\Xi;(v_2 \setminus f_2);\rho^{\smile};\kappa & \\
\sqsubseteq & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;\Phi^{\smile};v_0;\Xi;f_2;\lambda;\lambda^{\smile};\rho^{\smile};\kappa \sqcup v_0;\Xi;(v_2 \setminus f_2);\rho^{\smile};\kappa & \\
\sqsubseteq & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup v_0;\Xi;(v_2 \setminus f_2);\rho^{\smile};\kappa & \\
\sqsubseteq & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup v_0;\Xi;(v_2 \setminus f_2);\rho^{\smile};\pi;\pi^{\smile};\kappa & \\
= & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup v_0;\Xi;(v_2 \setminus f_2);\Xi^{\smile};v_0;\Phi;\pi^{\smile};\kappa & \text{tabulation} \\
\sqsubseteq & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup v_0;y_0;\Xi;(v_2 \setminus f_2);\Xi^{\smile};y_0;v_0;\Phi;\pi^{\smile};\kappa & \text{Defs. } f_2,\ y_0 \\
= & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup v_0;y_0;\Phi;\pi^{\smile};\kappa & \text{Lemma 4.1.7} \\
\sqsubseteq & u_0;\Xi;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup \Phi;X_0 & \\
\sqsubseteq & \Phi;\Phi^{\smile};u_0;\Xi;u_2;v_2;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup \Phi;X_0 & \\
\sqsubseteq & \Phi;\Phi^{\smile};u_0;\Xi;\lambda;\lambda^{\smile};u_2;v_2;\rho^{\smile};\kappa \sqcup \Phi;X_0;Q_2 \sqcup \Phi;X_0 & u_2;v_2 \sqsubseteq \mathsf{dom}\,\lambda \\
\sqsubseteq & \Phi;X_0;Q_2 \sqcup \Phi;X_0;Q_2 \sqcup \Phi;X_0 & \\
\sqsubseteq & \Phi;X_0;\Theta &
\end{array}
$$

$$
\begin{array}{llll}
u_0;\Xi;\lambda & \sqsubseteq & \Phi;c_1;\Phi^{\smile};u_0;\Xi;\lambda & u_0 \sqsubseteq \mathsf{dom}\,\Phi \text{ and } \mathsf{ran}\,(u_0;\Phi) \sqsubseteq c_1 \\
 & = & \Phi;\iota;\iota^{\smile};\Phi^{\smile};u_0;\Xi;\lambda & \\
 & = & \Phi;\iota;Q_3 \qquad \sqsubseteq \ \Phi;X_0;\Theta & \\
v_0;\Xi;\lambda & \sqsubseteq & \Phi;\Phi^{\smile};v_0;\Xi;\lambda & \mathsf{dom}\,(\Xi;k_2) \sqsubseteq \mathsf{dom}\,\Phi \\
 & = & \Phi;v_1;\Phi^{\smile};v_0;\Xi;\lambda & \\
 & = & \Phi;\pi^{\smile};\rho;\lambda & \text{tabulation} \\
 & = & \Phi;\pi^{\smile};\kappa;\kappa^{\smile};\rho;\lambda & \\
 & \sqsubseteq & \Phi;X_0;Q_2^{\smile} & \text{Def. } Q_2
\end{array}
$$

□

**Lemma B.4.3** X is univalent on $u_1 \sqcup v_1^{\smile}$.

**Proof**: With Lemma 6.1.3.i) we have $u_1 \sqcup v_1^{\smile} = c_1$, and with Lemma A.1.2.iii):

$$X^{\smile};c_1;X \sqsubseteq \mathbb{I} \Leftrightarrow \theta^{\smile};X_0^{\smile};c_1;X_0;\theta \sqsubseteq \mathbb{I} \Leftrightarrow X_0^{\smile};c_1;X_0 \sqsubseteq \theta;\theta^{\smile} \Leftrightarrow X_0^{\smile};c_1;X_0 \sqsubseteq \Theta$$

The left-hand side of the last inclusion expands to the following:

$$(\iota^{\smile} \sqcup \lambda^{\smile};\Xi^{\smile};\Phi \sqcup \kappa^{\smile};\pi);c_1;(\iota \sqcup \Phi^{\smile};\Xi;\lambda \sqcup \pi^{\smile};\kappa)$$

It therefore gives rise to nine inclusions in $\Theta$. Of these, six can be organised into symmetrical pairs, so we may omit one fromn each pair. For showing the rest, the key property is $c_1 = \iota;\iota^{\smile}$. Using this and the definitions of $Q_1$ and $Q_3$, we quickly obtain:

$$
\begin{array}{rcl}
\iota^{\smile};c_1;\iota & \sqsubseteq & \mathbb{I} \\
\iota^{\smile};c_1;\Phi^{\smile};\Xi;\lambda & \sqsubseteq & Q_3 \\
\iota^{\smile};c_1;\pi^{\smile};\kappa & = & Q_1 \\
\lambda^{\smile};\Xi^{\smile};\Phi;c_1;\Phi^{\smile};\Xi;\lambda & \sqsubseteq & Q_3^{\smile};Q_3 \\
\lambda^{\smile};\Xi^{\smile};\Phi;c_1;\pi^{\smile};\kappa & \sqsubseteq & Q_3^{\smile};Q_1 \\
\kappa^{\smile};\pi;c_1;\pi^{\smile};\kappa & = & Q_1^{\smile};Q_1
\end{array}
$$

□



**Lemma B.4.4** $\Psi$ is univalent on $(\mathsf{ran}\,\Xi)^{\smile}$.

**Proof**: As for X, the following needs to be contained in $\Theta$:

$$\Psi_0^{\smile};(\mathsf{ran}\,\Xi)^{\smile};\Psi_0 = (\iota^{\smile};\Phi^{\smile};u_0;\Xi \sqcup \kappa^{\smile};\rho \sqcup \lambda^{\smile});(\mathsf{ran}\,\Xi)^{\smile};(\Xi^{\smile};u_0;\Phi;\iota \sqcup \rho^{\smile};\kappa \sqcup \lambda)$$

We treat the six different constellations in the resulting nine-part join separately. Here, the key property is $(\mathsf{ran}\,\Xi)^{\smile} \sqsubseteq \lambda;\lambda^{\smile}$; its application yields:

$$
\begin{array}{rcl}
\iota^{\smile};\Phi^{\smile};u_0;\Xi;(\mathsf{ran}\,\Xi)^{\smile};\Xi^{\smile};u_0;\Phi;\iota & \sqsubseteq & Q_3;Q_3^{\smile} \\
\iota^{\smile};\Phi^{\smile};u_0;\Xi;(\mathsf{ran}\,\Xi)^{\smile};\rho^{\smile};\kappa & \sqsubseteq & Q_3;Q_2^{\smile} \\
\iota^{\smile};\Phi^{\smile};u_0;\Xi;(\mathsf{ran}\,\Xi)^{\smile};\lambda & \sqsubseteq & Q_3 \\
\kappa^{\smile};\rho;(\mathsf{ran}\,\Xi)^{\smile};\rho^{\smile};\kappa & \sqsubseteq & Q_2^{\smile};Q_2 \\
\kappa^{\smile};\rho;(\mathsf{ran}\,\Xi)^{\smile};\lambda & = & Q_2^{\smile} \\
\lambda^{\smile};(\mathsf{ran}\,\Xi)^{\smile};\lambda & \sqsubseteq & \mathbb{I}
\end{array}
$$

□

**Lemma B.4.5** Result coverage holds: $\mathsf{ran}\,(c_1;X) \sqcup \mathsf{ran}\,(w_2;\Psi) = \mathbb{I}$.

**Proof**:

$$
\begin{array}{lll}
 & \mathsf{ran}\,(c_1;X) \sqcup \mathsf{ran}\,(w_2;\Psi) & \\
= & \mathsf{ran}\,(c_1;X_0;\theta) \sqcup \mathsf{ran}\,(w_2;\Psi_0;\theta) & \\
= & \mathsf{ran}\,(\mathsf{ran}\,(c_1;X_0);\theta) \sqcup \mathsf{ran}\,(\mathsf{ran}\,(w_2;\Psi_0);\theta) & \\
= & \mathsf{ran}\,((\mathsf{ran}\,(c_1;X_0) \sqcup \mathsf{ran}\,(w_2;\Psi_0));\theta) & \\
\sqsupseteq & \mathsf{ran}\,((\mathsf{ran}\,(c_1;\iota) \sqcup \mathsf{ran}\,((v_2 \sqcup (\mathsf{ran}\,\Xi)^{\smile});\Psi_0));\theta) & \\
= & \mathsf{ran}\,((\mathsf{ran}\,\iota \sqcup \mathsf{ran}\,(v_2;\Psi_0 \sqcup (\mathsf{ran}\,\Xi)^{\smile};\Psi_0));\theta) & \text{Def. } \iota \\
= & \mathsf{ran}\,((\mathsf{ran}\,\iota \sqcup \mathsf{ran}\,((v_2 \setminus f_2);\Psi_0 \sqcup v_2;f_2;\Psi_0 \sqcup (\mathsf{ran}\,\Xi)^{\smile};\Psi_0));\theta) & \\
= & \mathsf{ran}\,((\mathsf{ran}\,\iota \sqcup \mathsf{ran}\,((v_2 \setminus f_2);\Psi_0) \sqcup \mathsf{ran}\,(k_2;\Psi_0));\theta) & \\
\sqsupseteq & \mathsf{ran}\,((\mathsf{ran}\,\iota \sqcup \mathsf{ran}\,((v_2 \setminus f_2);\rho^{\smile};\kappa) \sqcup \mathsf{ran}\,(k_2;\lambda));\theta) & \\
= & \mathsf{ran}\,((\mathsf{ran}\,\iota \sqcup \mathsf{ran}\,\kappa \sqcup \mathsf{ran}\,\lambda);\theta) & \text{Def. } \lambda \\
= & \mathsf{ran}\,\theta & \\
= & \mathbb{I} &
\end{array}
$$

□

**Lemma B.4.6** Parameter tabulation holds: $\Psi^{\smile};v_2;\Psi \sqcap X^{\smile};v_1;X = v_3$.

**Proof**: Since

$$
\begin{array}{lll}
v_3 & = & \mathsf{ran}\,(v_1;X) \ = \ \mathsf{ran}\,(v_1;X) \sqcap \mathsf{ran}\,(\mathsf{ran}\,(v_0;\Phi);X) \\
 & = & \mathsf{ran}\,(v_1;X) \sqcap \mathsf{ran}\,(v_0;\Phi;X) \ = \ \mathsf{ran}\,(v_1;X) \sqcap \mathsf{ran}\,(v_0;\Xi;\Psi) \\
 & = & \mathsf{ran}\,(v_1;X) \sqcap \mathsf{ran}\,(\mathsf{ran}\,(v_0;\Xi);\Psi) \ = \ \mathsf{ran}\,(v_1;X) \sqcap \mathsf{ran}\,(v_2;\Psi) \ \sqsubseteq \ \Psi^{\smile};v_2;\Psi \sqcap X^{\smile};v_1;X\ ,
\end{array}
$$

we only need to show $\Psi^{\smile};v_2;\Psi \sqcap X^{\smile};v_1;X \sqsubseteq \mathbb{I}$. We have:

$$
\begin{array}{lll}
v_2;\Psi_0 & = & v_2;\Xi^{\smile};u_0;\Phi;\iota \sqcup v_2;\rho^{\smile};\kappa \sqcup v_2;\lambda \\
 & = & (v_2 \sqcap u_2);\Xi^{\smile};u_0;\Phi;\iota \sqcup (v_2 \setminus f_2);\rho^{\smile};\kappa \sqcup v_2;f_2;\lambda \\
 & = & b_2;\Xi^{\smile};u_0;\Phi;\iota \sqcup (v_2 \setminus f_2);\rho^{\smile};\kappa \sqcup v_2;f_2;\lambda
\end{array}
$$

For $\Psi_0^{\smile};v_2;\Psi_0$ we then have the following six components, and the converses of the three mixed terms:

$$\begin{array}{rcl}
\iota^{\smile};\Phi^{\smile};u_0;\Xi;b_2;\Xi^{\smile};u_0;\Phi;\iota & & \\
\iota^{\smile};\Phi^{\smile};u_0;\Xi;b_2;(v_2 \setminus f_2);\rho^{\smile};\kappa & & \\
\iota^{\smile};\Phi^{\smile};u_0;\Xi;b_2;v_2;f_2;\lambda & & \\
\kappa^{\smile};\rho;(v_2 \setminus f_2);\rho^{\smile};\kappa & \sqsubseteq & \kappa^{\smile};\rho;\rho^{\smile};\kappa \\
\kappa^{\smile};\rho;(v_2 \setminus f_2);v_2;f_2;\lambda & \sqsubseteq & Q_2^{\smile} \\
\lambda^{\smile};v_2;f_2;\lambda & = & \mathsf{ran}\,(v_2;\lambda) \ \sqsubseteq \ \mathbb{I}
\end{array}$$

Analogous preparation for $\mathrm{X}_0$:

$$v_1;\mathrm{X}_0 = v_1;\iota \sqcup v_1;\pi^{\smile};\kappa \sqcup v_1;\Phi^{\smile};\Xi;\lambda = b_1;\iota \sqcup \pi^{\smile};\kappa \sqcup v_1;\Phi^{\smile};\Xi;\lambda$$

For $\mathrm{X}_0^{\smile};v_1;\mathrm{X}_0$ we then have the following terms corresponding to those above:

$$\begin{array}{rcl}
\iota^{\smile};b_1;\iota & \sqsubseteq & \mathbb{I} \\
\iota^{\smile};b_1;\pi^{\smile};\kappa & \sqsubseteq & Q_1 \\
\iota^{\smile};b_1;v_1;\Phi^{\smile};\Xi;\lambda & = & \iota^{\smile};b_1;\Phi^{\smile};\Xi;\lambda \ \sqsubseteq \ Q_3 \\
\kappa^{\smile};\pi;\pi^{\smile};\kappa & & \\
\kappa^{\smile};\pi;v_1;\Phi^{\smile};\Xi;\lambda & & \\
\lambda^{\smile};\Xi^{\smile};\Phi;v_1;\Phi^{\smile};\Xi;\lambda & &
\end{array}$$

Since intersections of terms with different injection on either end are empty, we obtain the following for the intersection of the two lists:

$$\begin{array}{rcl}
\Psi_0^{\smile};v_2;\Psi_0 \sqcap \mathrm{X}_0^{\smile};v_1;\mathrm{X}_0 & \sqsubseteq & \mathbb{I} \sqcup Q_1 \sqcup Q_3 \sqcup (\kappa^{\smile};\rho;\rho^{\smile};\kappa \sqcap \kappa^{\smile};\pi;\pi^{\smile};\kappa) \sqcup Q_2^{\smile} \sqcup \mathbb{I} \\
& \sqsubseteq & \Theta \sqcup \kappa^{\smile};(\rho;\rho^{\smile} \sqcap \pi;\pi^{\smile});\kappa \\
& \sqsubseteq & \Theta
\end{array}$$

All this together now allows us to show the desired inclusion:

$$\begin{array}{rll}
& \Psi^{\smile};v_2;\Psi \sqcap \mathrm{X}^{\smile};v_1;\mathrm{X} & \\
= & \theta^{\smile};\Psi_0^{\smile};v_2;\Psi_0;\theta \sqcap \theta^{\smile};\mathrm{X}_0^{\smile};v_1;\mathrm{X}_0;\theta & \\
= & \theta^{\smile};(\Theta;\Psi_0^{\smile};v_2;\Psi_0;\Theta \sqcap \mathrm{X}_0^{\smile};v_1;\mathrm{X}_0);\theta & \theta \text{ univalent} \\
\sqsubseteq & \theta^{\smile};((\Psi_0^{\smile};v_2;\Psi_0 \sqcup \Theta;\Psi_0^{\smile};b_2;\Psi_0;\Theta) \sqcap \mathrm{X}_0^{\smile};v_1;\mathrm{X}_0);\theta & \\
= & \theta^{\smile};((\Psi_0^{\smile};v_2;\Psi_0 \sqcup \theta;\Psi^{\smile};b_2;\Psi;\theta^{\smile}) \sqcap \mathrm{X}_0^{\smile};v_1;\mathrm{X}_0);\theta & \\
\sqsubseteq & \theta^{\smile};((\Psi_0^{\smile};v_2;\Psi_0 \sqcup \Theta) \sqcap \mathrm{X}_0^{\smile};v_1;\mathrm{X}_0);\theta & \Psi \text{ univalent on } b_2 \\
\sqsubseteq & \theta^{\smile};((\Psi_0^{\smile};v_2;\Psi_0 \sqcap \mathrm{X}_0^{\smile};v_1;\mathrm{X}_0) \sqcup \Theta);\theta & \\
\sqsubseteq & \theta^{\smile};\Theta;\theta & \\
\sqsubseteq & \theta^{\smile};\theta;\theta^{\smile};\theta & \Theta = \theta;\theta^{\smile} \\
\sqsubseteq & \mathbb{I} & \theta \text{ univalent}
\end{array}$$

$\square$

The last four lemmata together imply the combination property.



While interface preservation lets interface components $u$ propagate very freely, the condition (spurious) for reasonable gluing setups (Def. 6.5.4) lets $f_0$ propagate at least under certain circumstances:

**Lemma B.4.7** [←198, 199] $(\Phi^{\smile};f_0;\Xi);(\Xi^{\smile};u_0;\Phi) = (\Phi^{\smile};u_0;f_0;\Xi);(\Xi^{\smile};u_0;f_0;\Phi)$

**Proof**: As preparation, we first show a more general equation:

$$\begin{array}{rcll}
(\Phi^{\smile};f_0;\Xi);\Xi^{\smile};\Phi & = & \Phi^{\smile};f_0;\Xi;\Xi^{\smile};f_0;\Phi \sqcup \Phi^{\smile};f_0;\Xi;\Xi^{\smile};y_0;\Phi & (\mathsf{spurious}) \\
& = & \Phi^{\smile};f_0;\Xi;\Xi^{\smile};f_0;\Phi \sqcup \Phi^{\smile};f_0;\Xi;\Xi^{\smile};f_0;y_0;\Phi & \Xi;\Xi^{\smile};y_0 \sqsubseteq \mathbb{I} \\
& = & \Phi^{\smile};f_0;\Xi;\Xi^{\smile};f_0;\Phi & \\
(\Phi^{\smile};f_0;\Xi);(\Xi^{\smile};u_0;\Phi) & = & (\Phi^{\smile};f_0;\Xi);\Xi^{\smile};\Phi;u_1 & \\
& = & \Phi^{\smile};f_0;\Xi;\Xi^{\smile};f_0;\Phi;u_1 & \text{above} \\
& = & \Phi^{\smile};f_0;\Xi;\Xi^{\smile};f_0;u_0;\Phi & \\
& = & \Phi^{\smile};f_0;\Xi;u_2;\Xi^{\smile};u_0;f_0;\Phi & \\
& = & \Phi^{\smile};u_0;f_0;\Xi;\Xi^{\smile};u_0;f_0;\Phi &
\end{array}$$

$\square$

**Lemma B.4.8** [←196, 198] $\mathrm{X}_0;(Q \sqcup Q^{\smile}) \sqsubseteq \mathrm{X}_0;(\mathbb{I} \sqcup Q_2 \sqcup Q_4) \sqsubseteq \mathrm{X}_0 \sqcup \Phi^{\smile};f_0;\Xi;(u_2 \sqcup f_2;(v_2 \setminus f_2));\Psi_0$

**Proof**:

$$\begin{array}{rcl}
\mathrm{X}_0;Q_1 & = & \iota;Q_1 \ = \ \iota;\iota^{\smile};\pi^{\smile};\kappa \ = \ c_1;\pi^{\smile};\kappa \ \sqsubseteq \ \mathrm{X}_0 \\
\mathrm{X}_0;Q_3 & = & \iota;Q_3 \ = \ \iota;\iota^{\smile};\Phi^{\smile};\Xi;\lambda \ = \ c_1;\Phi^{\smile};\Xi;\lambda \ \sqsubseteq \ \mathrm{X}_0 \\
\mathrm{X}_0;Q_4 & = & \iota;Q_4 \ = \ \iota;\iota^{\smile};\Phi^{\smile};y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};\Phi;\iota \\
& = & c_1;\Phi^{\smile};y_0^{\sim};\Xi;\Xi^{\smile};y_0^{\sim};\Phi;\iota \ \sqsubseteq \ \Phi^{\smile};u_0;f_0;\Xi;u_2;\Psi_0 \\
\mathrm{X}_0;Q_2 & = & \Phi^{\smile};\Xi;\lambda;Q_2 \ = \ \Phi^{\smile};\Xi;\lambda;\lambda^{\smile};\rho^{\smile};\kappa \\
& = & \Phi^{\smile};\Xi;k_2;(v_2 \setminus f_2);\rho^{\smile};\kappa \\
& = & \quad \{\ k_2 \sqcap v_2 \sqsubseteq (v_2 \sqcap f_2) \sqcup b_2 \sqsubseteq f_2 \ \} \\
& & \Phi^{\smile};\Xi;f_2;(v_2 \setminus f_2);\Psi_0 \ = \ \Phi^{\smile};f_0;\Xi;f_2;(v_2 \setminus f_2);\Psi_0 \\
\mathrm{X}_0;Q_3^{\smile} & = & \Phi^{\smile};\Xi;\lambda;Q_3^{\smile} \ = \ \Phi^{\smile};\Xi;\lambda;\lambda^{\smile};\Xi^{\smile};\Phi;\iota \\
& = & \Phi^{\smile};\Xi;k_2;\Xi^{\smile};\Phi;\iota \ = \ \Phi^{\smile};\Xi;k_2;\Xi^{\smile};u_0;\Phi;\iota \\
& = & \Phi^{\smile};(y_0 \sqcap u_0 \sqcap \Xi;k_2;\Xi^{\smile});\Phi;\iota \sqcup \Phi^{\smile};y_0^{\sim};\Xi\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota \\
& \sqsubseteq & \Phi^{\smile};\mathsf{dom}\,(\mathsf{upa}\,\Phi);\Phi;\iota \sqcup \Phi^{\smile};y_0^{\sim};\Xi;u_2\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota \\
& \sqsubseteq & \iota \sqcup \Phi^{\smile};u_0;y_0^{\sim};\Xi;u_2\Xi^{\smile};y_0^{\sim};u_0;\Phi;\iota \ \sqsubseteq \ \mathrm{X}_0 \sqcup u_2;\Phi^{\smile};y_0^{\sim};\Xi\Xi^{\smile};y_0^{\sim};\Phi;\iota \\
& \sqsubseteq & \mathrm{X}_0 \sqcup \iota;\iota^{\smile};\Phi^{\smile};y_0^{\sim};\Xi\Xi^{\smile};y_0^{\sim};\Phi;\iota \ \sqsubseteq \ \mathrm{X}_0 \sqcup \mathrm{X}_0;Q_4 \\
\mathrm{X}_0;Q_1^{\smile} & = & \pi^{\smile};\kappa;Q_1^{\smile} \ = \ \pi^{\smile};\kappa;\kappa^{\smile};\pi;\iota \ = \ \pi^{\smile};\pi;\iota \ \sqsubseteq \ \iota \ \sqsubseteq \ \mathrm{X}_0 \\
\mathrm{X}_0;Q_2^{\smile} & = & \pi^{\smile};\kappa;Q_2^{\smile} \ = \ \pi^{\smile};\kappa;\kappa^{\smile};\rho;\lambda \\
& = & \pi^{\smile};\rho;\lambda \ = \ \Phi^{\smile};v_0;\Xi;\lambda \ \sqsubseteq \ \mathrm{X}_0
\end{array}$$

$\square$

**Lemma B.4.9** [←196, 198] $\Psi_0;(Q \sqcup Q^\smile) \sqsubseteq (v_2 \sqcup u_2;k_2);\Psi_0 \sqcup \Xi^\smile;u_0;\Phi;\mathrm{X}_0;(\mathbb{I} \sqcup Q_4)$

**Proof:**

$$\begin{aligned}
\Psi_0;Q_1 &= \Xi^\smile;u_0;\Phi;\iota;Q_1 \;=\; \Xi^\smile;u_0;\Phi;\iota;\iota^\smile;\pi^\smile;\kappa\\
&= \Xi^\smile;u_0;\Phi;c_1;\pi^\smile;\kappa \;=\; \Xi^\smile;u_0;\Phi;c_1;v_1;\pi^\smile;\kappa\\
&= \Xi^\smile;u_0;\Phi;b_1;\pi^\smile;\kappa \;=\; \Xi^\smile;u_0;f_0;\Phi;b_1;\pi^\smile;\kappa \;\sqsubseteq\; \Xi^\smile;u_0;f_0;\Phi;\mathrm{X}_0\\
\Psi_0;Q_3 &= \Xi^\smile;u_0;\Phi;\iota;Q_3 \;=\; \Xi^\smile;u_0;\Phi;\iota;\iota^\smile;\Phi^\smile;\Xi;\lambda\\
&= \Xi^\smile;u_0;\Phi;c_1;\Phi^\smile;\Xi;\lambda \;\sqsubseteq\; \Xi^\smile;u_0;\Phi;\mathrm{X}_0\\
\Psi_0;Q_4 &= \Xi^\smile;u_0;\Phi;\iota;Q_4 \;\sqsubseteq\; \Xi^\smile;u_0;\Phi;\mathrm{X}_0;Q_4\\
\Psi_0;Q_1^\smile &= \rho^\smile;\kappa;Q_1^\smile \;=\; \rho^\smile;\kappa;\kappa^\smile;\pi;\iota\\
&= \rho^\smile;\pi;\iota \;=\; \Xi^\smile;v_0;\Phi;\iota \;\sqsubseteq\; v_2;\Psi_0\\
\Psi_0;Q_2^\smile &= \rho^\smile;\kappa;Q_2^\smile \;=\; \rho^\smile;\kappa;\kappa^\smile;\rho;\lambda\\
&= \rho^\smile;\rho;\lambda \;\sqsubseteq\; v_2;\lambda \;\sqsubseteq\; v_2;\Psi_0\\
\Psi_0;Q_2 &= \lambda;Q_2 \;=\; \lambda;\lambda^\smile;\rho^\smile;\kappa \;=\; k_2;\rho^\smile;\kappa \;\sqsubseteq\; k_2;v_2;\Psi_0\\
\Psi_0;Q_3^\smile &= \lambda;Q_3^\smile \;=\; \lambda;\lambda^\smile;\Xi^\smile;\Phi;\iota \;=\; k_2;\Xi^\smile;u_0;\Phi;\iota \;\sqsubseteq\; k_2;u_2;\Psi_0
\end{aligned}$$

□

**Lemma B.4.10** $\mathrm{X}_0;\Theta = \mathrm{X}_0 \sqcup (\Phi^\smile;f_0;\Xi);\Psi_0;\Theta$ and $\Psi_0;\Theta = \Psi_0 \sqcup (\Xi^\smile;u_0;\Phi);\mathrm{X}_0;\Theta$.

**Proof**: Via cyclic inclusion chains, using the definitions of X, $\Psi$, and $\Theta$, Lemma B.4.8 and Lemma B.4.9, commutativity, and univalence of $\Phi$ on $f_0$ and of $\Xi$ on $u_0$:

$$\begin{aligned}
\mathrm{X}_0;\Theta &\sqsubseteq \mathrm{X}_0 \sqcup \Phi^\smile;f_0;\Xi;\Psi_0;\Theta \;=\; \mathrm{X}_0 \sqcup \Phi^\smile;f_0;\Xi;\Psi;\theta^\smile\\
&= \mathrm{X}_0 \sqcup \Phi^\smile;f_0;\Phi;\mathrm{X};\theta^\smile \;\sqsubseteq\; \mathrm{X}_0 \sqcup \mathrm{X};\theta^\smile \;=\; \mathrm{X}_0 \sqcup \mathrm{X}_0;\Theta \;=\; \mathrm{X}_0;\Theta\\
\Psi_0;\Theta &\sqsubseteq \Psi_0 \sqcup \Xi^\smile;u_0;\Phi;\mathrm{X}_0;\Theta \;=\; \Psi_0 \sqcup \Xi^\smile;u_0;\Phi;\mathrm{X};\theta^\smile\\
&= \Psi_0 \sqcup \Xi^\smile;u_0;\Xi;\Psi;\theta^\smile \;\sqsubseteq\; \Psi_0 \sqcup \Psi;\theta^\smile \;=\; \Psi_0 \sqcup \Psi_0;\Theta \;=\; \Psi_0;\Theta
\end{aligned}$$

□

Let us define partial identities for the *used parameter parts*:

$$v_1' := \mathsf{dom}\,(\Phi^\smile;v_0;\Xi) \qquad \text{and} \qquad v_2' := \mathsf{ran}\,(\Phi^\smile;v_0;\Xi)\ .$$

In preparation for the alternative commutativity properties we first calculate the respective compositions of $\mathrm{X}_0$ and $\Psi_0$:

**Lemma B.4.11** [←198, 199]

$$\begin{aligned}
\mathrm{X}_0;\mathrm{X}_0^\smile &= c_1 \sqcup v_1' \sqcup \Phi^\smile;f_0;\Xi;k_2;\Xi^\smile;f_0;\Phi\\
\Psi_0;\Psi_0^\smile &= h_2 \sqcup v_2' \sqcup \Xi^\smile;u_0;\Phi;\Phi^\smile;u_0;\Xi\\
\mathrm{X}_0;\Psi_0^\smile &= \Phi^\smile;\Xi
\end{aligned}$$

**Proof**:

$$\begin{aligned}
\mathrm{X}_0;\mathrm{X}_0^\smile &= \iota;\iota^\smile \sqcup \pi^\smile;\kappa;\kappa^\smile;\pi \sqcup \Phi^\smile;\Xi;\lambda;\lambda^\smile;\Xi^\smile;\Phi\\
&= c_1 \sqcup \pi^\smile;\pi \sqcup \Phi^\smile;\Xi;k_2;\Xi^\smile;\Phi\\
&= c_1 \sqcup \mathsf{dom}\,(\Phi^\smile;v_0;\Xi) \sqcup \Phi^\smile;f_0;\Xi;k_2;\Xi^\smile;f_0;\Phi && k_2 \sqcap \mathsf{ran}\,\Xi \sqsubseteq f_2\\
&= c_1 \sqcup v_1' \sqcup \Phi^\smile;f_0;\Xi;k_2;\Xi^\smile;f_0;\Phi
\end{aligned}$$



$$\begin{aligned}
\Psi_0;\Psi_0^\smile &= (\Xi^\smile;\Phi;\iota \sqcup \rho^\smile;\kappa \sqcup \lambda);(\iota^\smile;\Phi^\smile;\Xi \sqcup \kappa^\smile;\rho \sqcup \lambda^\smile)\\
&= \Xi^\smile;\Phi;\iota;\iota^\smile;\Phi^\smile;\Xi \sqcup \rho^\smile;\kappa;\kappa^\smile;\rho \sqcup \lambda;\gamma\\
&= \Xi^\smile;\Phi;c_1;\Phi^\smile;\Xi \sqcup \rho^\smile;\rho \sqcup k_2\\
&= \Xi^\smile;\Phi;c_1;\Phi^\smile;\Xi \sqcup \mathsf{ran}\,W \sqcup k_2\\
&= \Xi^\smile;\Phi;c_1;\Phi^\smile;\Xi \sqcup v_2' \sqcup h_2 \sqcup b_2\\
&= \Xi^\smile;\Phi;c_1;\Phi^\smile;\Xi \sqcup v_2' \sqcup h_2 && b_2 \sqsubseteq \mathsf{ran}\,(c_1;\Phi^\smile;\Xi)\\
&= h_2 \sqcup v_2' \sqcup \Xi^\smile;u_0;\Phi;\Phi^\smile;u_0;\Xi\\
\mathrm{X}_0;\Psi_0^\smile &= (\iota \sqcup \pi^\smile;\kappa \sqcup \Phi^\smile;\Xi;\lambda);(\iota^\smile;\Phi^\smile;\Xi \sqcup \kappa^\smile;\rho \sqcup \lambda^\smile)\\
&= \iota;\iota^\smile;\Phi^\smile;\Xi \sqcup \pi^\smile;\kappa;\kappa^\smile;\rho \sqcup \Phi^\smile;\Xi;\lambda;\lambda^\smile\\
&= c_1;\Phi^\smile;\Xi \sqcup \pi^\smile;\rho \sqcup \Phi^\smile;\Xi;k_2\\
&= c_1;\Phi^\smile;\Xi \sqcup v_1;\Phi^\smile;v_0;\Xi;(v_2 \setminus f_2) \sqcup \Phi^\smile;\Xi;k_2 && \text{tabulation, } k_2 \sqcap \mathsf{ran}\,\Xi \sqsubseteq f_2\\
&= \Phi^\smile;\Xi
\end{aligned}$$

□

We may now use these equations for deriving relations between $\mathrm{X};\Psi^\smile$ and $\mathrm{X};\mathrm{X}^\smile$ and $\Psi;\Psi^\smile$:

$$\begin{aligned}
\mathrm{X};\Psi^\smile &= \mathrm{X}_0;\Theta;\Psi_0^\smile \;=\; (\mathrm{X}_0 \sqcup \Phi^\smile;f_0;\Xi;\Psi_0;\Theta);\Psi_0^\smile\\
&= \mathrm{X}_0;\Psi_0^\smile \sqcup \Phi^\smile;f_0;\Xi;\Psi_0;\Theta;\Psi_0^\smile \;=\; \Phi^\smile;\Xi \sqcup \Phi^\smile;f_0;\Xi;\Psi;\Psi^\smile
\end{aligned}$$

$$\begin{aligned}
\mathrm{X};\mathrm{X}^\smile &= \mathrm{X}_0;\Theta;\mathrm{X}_0^\smile\\
&= \mathrm{X}_0;(\mathrm{X}_0^\smile \sqcup \Theta;\Psi_0^\smile;\Xi^\smile;f_0;\Phi)\\
&= \mathrm{X}_0;\mathrm{X}_0^\smile \sqcup \mathrm{X}_0;\Theta;\Psi_0^\smile;\Xi^\smile;f_0;\Phi)\\
&= c_1 \sqcup v_1' \sqcup (\Phi^\smile;f_0;\Xi);k_2;(\Xi^\smile;f_0;\Phi) \sqcup \mathrm{X};\Psi^\smile;(\Xi^\smile;f_0;\Phi)\\
&= c_1 \sqcup v_1' \sqcup ((\Phi^\smile;f_0;\Xi);k_2 \sqcup \mathrm{X};\Psi^\smile);(\Xi^\smile;f_0;\Phi)\\
&= c_1 \sqcup v_1' \sqcup \mathrm{X};\Psi^\smile;(\Xi^\smile;f_0;\Phi) && \Phi^\smile;f_0;\Xi;k_2 \sqsubseteq \mathrm{X};\Psi^\smile
\end{aligned}$$

$$\begin{aligned}
\Psi;\Psi^\smile &= \Psi_0;\Theta;\Psi_0^\smile\\
&= \Psi_0;\Psi_0^\smile \sqcup \Xi^\smile;u_0;\Phi;\mathrm{X}_0;\Theta;\Psi_0^\smile\\
&= \Psi_0;\Psi_0^\smile \sqcup \Xi^\smile;u_0;\Phi;\mathrm{X};\Psi^\smile\\
&= h_2 \sqcup v_2' \sqcup \Xi^\smile;u_0;\Phi;\Phi^\smile;\Xi \sqcup \Xi^\smile;u_0;\Phi;\mathrm{X};\Psi^\smile\\
&= h_2 \sqcup v_2' \sqcup (\Xi^\smile;u_0;\Phi);((\Phi^\smile;u_0;\Xi) \sqcup \mathrm{X};\Psi^\smile)\\
&= h_2 \sqcup v_2' \sqcup (\Xi^\smile;u_0;\Phi);\mathrm{X};\Psi^\smile && \Phi^\smile;u_0;\Xi \sqsubseteq \Phi^\smile;\Xi \sqsubseteq \mathrm{X};\Psi^\smile
\end{aligned}$$

For alternative commutativity we now may continue:

$$\begin{aligned}
\mathrm{X};\Psi^\smile &= \Phi^\smile;\Xi \sqcup (\Phi^\smile;f_0;\Xi);\Psi;\Psi^\smile\\
&= \Phi^\smile;\Xi \sqcup (\Phi^\smile;f_0;\Xi);(h_2 \sqcup v_2' \sqcup (\Xi^\smile;u_0;\Phi);\mathrm{X};\Psi^\smile)\\
&= \Phi^\smile;\Xi \sqcup (\Phi^\smile;f_0;\Xi);(h_2 \sqcup v_2') \sqcup (\Phi^\smile;f_0;\Xi);(\Xi^\smile;u_0;\Phi);\mathrm{X};\Psi^\smile\\
&= \Phi^\smile;\Xi \sqcup (\Phi^\smile;f_0;\Xi);(\Xi^\smile;u_0;\Phi);\mathrm{X};\Psi^\smile
\end{aligned}$$

Altogether we have now shown the following (mutually and directly) recursive equations for $\mathrm{X};\Psi^\smile$ and $\mathrm{X};\mathrm{X}^\smile$ and $\Psi;\Psi^\smile$ (for the directly recursive equation for $\mathrm{X};\mathrm{X}^\smile$ we used an alternative presentation for $\mathrm{X};\Psi^\smile$ that is obtained via resolving $\Theta;\Psi^\smile$ first):

**Lemma B.4.12**

$$\begin{array}{rcl}
\mathrm{X};\Psi^{\smile} &=& \Phi^{\smile};\Xi \sqcup \Phi^{\smile};f_0;\Xi;\Psi;\Psi^{\smile}\\
&=& \Phi^{\smile};\Xi \sqcup (\Phi^{\smile};f_0;\Xi);(\Xi^{\smile};u_0;\Phi);\mathrm{X};\Psi^{\smile}\\
\mathrm{X};\mathrm{X}^{\smile} &=& c_1 \sqcup v_1' \sqcup \mathrm{X};\Psi^{\smile};(\Xi^{\smile};f_0;\Phi)\\
&=& c_1 \sqcup v_1' \sqcup (\Phi^{\smile};\Xi \sqcup \mathrm{X};\mathrm{X}^{\smile};(\Phi^{\smile};u_0;\Xi));(\Xi^{\smile};f_0;\Phi)\\
\Psi;\Psi^{\smile} &=& h_2 \sqcup v_2' \sqcup (\Xi^{\smile};u_0;\Phi);\mathrm{X};\Psi^{\smile}\\
&=& h_2 \sqcup v_2' \sqcup (\Xi^{\smile};u_0;\Phi);((\Phi^{\smile};u_0;\Xi) \sqcup (\Phi^{\smile};u_0;f_0;\Xi);\Psi;\Psi^{\smile})
\end{array}$$ □

This shows, by a standard argument for such recursive equations (in fact, only $\sqsupseteq$ is needed), one inclusion for the alternative commutativity conditions for weak pullouts (we also use Lemma B.4.7 to be able to further abbreviate the reflexive transitive closures):

$$\begin{array}{rcl}
\mathrm{X};\Psi^{\smile} &\sqsupseteq& (\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};\Phi^{\smile};\Xi\\
\mathrm{X};\mathrm{X}^{\smile} &\sqsupseteq& c_1 \sqcup v_1' \sqcup (\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};(\Phi^{\smile};f_0;\Xi);(\Xi^{\smile};f_0;\Phi)\\
\Psi;\Psi^{\smile} &\sqsupseteq& h_2 \sqcup v_2' \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};(\Phi^{\smile};u_0;\Xi)
\end{array}$$

We are going to show the opposite inclusion only for $\Psi;\Psi^{\smile}$; the other equations then follow via Lemma B.4.12.

For this, we continue the inclusion of Lemma B.4.9 towards a "tail-recursive" shape:

**Lemma B.4.13** [←199] $\Psi_0;(Q \sqcup Q^{\smile}) \sqsubseteq \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi);\Psi_0$

**Proof**: First an auxiliary calculation:

$$\begin{array}{rll}
& (\Xi^{\smile};u_0;\Phi);\mathrm{X}_0 &\\
= & (\Xi^{\smile};u_0;\Phi);\iota \sqcup (\Xi^{\smile};u_0;\Phi);\pi^{\smile};\kappa \sqcup (\Xi^{\smile};u_0;\Phi);\Phi^{\smile};\Xi;\lambda &\\
\sqsubseteq & \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);\pi^{\smile};\rho;\rho^{\smile};\kappa \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};f_0;\Xi);\Psi_0 & \rho \text{ total, Def. } \lambda,\ \Psi_0\\
\sqsubseteq & \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};v_0;\Xi);\rho^{\smile};\kappa \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};f_0;\Xi);\Psi_0 & \text{tabulation}\\
= & \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};b_0;\Xi);\rho^{\smile};\kappa \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};f_0;\Xi);\Psi_0 & \text{interf. pres.}\\
= & \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};f_0;\Xi);\Psi_0 & \text{interf. pres.}
\end{array}$$

This allows us to conclude:

$$\begin{array}{rll}
& \Psi_0;(Q \sqcup Q^{\smile}) &\\
\sqsubseteq & \Psi_0 \sqcup \Xi^{\smile};u_0;\Phi;\mathrm{X}_0;(\mathbb{I} \sqcup Q_4) & \text{Lemma B.4.9}\\
= & \Psi_0 \sqcup \Xi^{\smile};u_0;\Phi;\mathrm{X}_0 \sqcup \Xi^{\smile};u_0;\Phi;\mathrm{X}_0;Q_4 &\\
\sqsubseteq & \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};f_0;\Xi);\Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi);\Psi_0 & \text{Lemma B.4.8}\\
= & \Psi_0 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi);\Psi_0 &
\end{array}$$ □

Now we are equipped to show the desired inclusion:

**Lemma B.4.14** $\Psi;\Psi^{\smile} \sqsubseteq v_2' \sqcup h_2 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};(\Phi^{\smile};u_0;\Xi)$

**Proof**: From Lemma B.4.11 we know that the statement is true when ignoring the $\Theta$-component of $\Psi;\Psi^{\smile} = \Psi_0;\Theta;\Psi_0^{\smile}$. Standard arguments over the reflexive transitive closure



then allow us to conclude:

$$\begin{array}{rcll}
\Psi;\Psi^{\smile} &=& \Psi_0;(Q \sqcup Q^{\smile})^{*};\Psi_0^{\smile} &\\
&\sqsubseteq& ((\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi))^{*};\Psi_0;\Psi_0^{\smile} & \text{Lemma B.4.13}\\
&\sqsubseteq& ((\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi))^{*};(v_2' \sqcup h_2 \sqcup \Xi^{\smile};u_0;\Phi;\Phi^{\smile};u_0;\Xi) & \text{Lemma B.4.11}\\
&=& v_2' \sqcup h_2 \sqcup ((\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi))^{*};(\Xi^{\smile};u_0;\Phi;\Phi^{\smile};u_0;\Xi) &\\
&=& v_2' \sqcup h_2 \sqcup (\Xi^{\smile};u_0;\Phi);((\Phi^{\smile};u_0;f_0;\Xi);(\Xi^{\smile};u_0;\Phi))^{*};(\Phi^{\smile};u_0;\Xi) &\\
&=& v_2' \sqcup h_2 \sqcup (\Xi^{\smile};u_0;\Phi);(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};(\Phi^{\smile};u_0;\Xi) & \text{Lemma B.4.7}
\end{array}$$ □

With this inclusion, we have shown the equation for $\Psi;\Psi^{\smile}$ from the weak pullout definition. With the equations of Lemma B.4.12 we also abtain the equations for $\mathrm{X};\Psi^{\smile}$ and $\mathrm{X};\mathrm{X}^{\smile}$, and this finishes the proof of correctness of the direct result construction.

## B.5 The Straight Host Construction Yields Weak Pullout Complements

**Proof of Theorem 6.5.9**: Since $u_0;\Phi$ is univalent, we have $f_0 \sqsubseteq u_0$.

$$\begin{array}{rcll}
\Phi^{\smile};v_0;\Xi &=& \Phi^{\smile};v_0;\Phi;\mathrm{X};\Psi^{\smile} \ = \ v_1;\mathrm{X};\Psi^{\smile} &\\
\Phi^{\smile};u_0;f_0;\Xi &=& \Phi^{\smile};u_0;\Phi;\mathrm{X};\Psi^{\smile} \ = \ u_1;\mathrm{X};\Psi^{\smile} &\\
\Phi^{\smile};u_0;f_0;\Xi;\Xi^{\smile};u_0;f_0;\Phi &=& u_1;\mathrm{X};\Psi^{\smile};\Psi;\mathrm{X}^{\smile};u_1 \ = \ u_1;\mathrm{X};\mathrm{X}^{\smile};u_1 & \mathsf{ran}\,(u_1;\mathrm{X}) \sqsubseteq \mathsf{ran}\,\Psi\\
\Xi^{\smile};u_0;f_0;\Phi;\Phi^{\smile};f_0;u_0;\Xi &=& \Psi;\mathrm{X}^{\smile};u_1;\mathrm{X};\Psi^{\smile} \ \sqsubseteq \ \Psi;\Psi^{\smile} \ = \ \mathbb{I} & u_1;\mathrm{X} \text{ univalent}
\end{array}$$

The last two lines of this show the following:

$$\begin{array}{rcl}
(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleleft} &=& \mathbb{I}\\
(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright} &=& \mathbb{I} \sqcup \Phi^{\smile};u_0;f_0;\Xi;(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleleft};\Xi^{\smile};u_0;f_0;\Phi\\
&=& \mathbb{I} \sqcup \Phi^{\smile};u_0;f_0;\Xi;;\Xi^{\smile};u_0;f_0;\Phi\\
&=& \mathbb{I} \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1
\end{array}$$

Then:

$$\begin{array}{rcll}
(\Phi^{\smile};u_0;f_0;\Xi)^{\triangleright};\Phi^{\smile};\Xi &=& \Phi^{\smile};\Xi \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1;\Phi^{\smile};\Xi &\\
&=& \Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile} \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1;\Phi^{\smile};\Phi;\mathrm{X};\Psi^{\smile} &\\
&=& \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} \sqcup u_1;\mathrm{X};\mathrm{X}^{\smile};u_1;\mathrm{X};\Psi^{\smile} & u_1 \sqsubseteq \mathsf{ran}\,\Phi\\
&=& \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} \sqcup u_1;\mathrm{X};\Psi^{\smile} & u_1;\mathrm{X} \text{ univalent}\\
&=& \mathsf{ran}\,\Phi;\mathrm{X};\Psi^{\smile} &\\
&=& \mathrm{X};\Psi^{\smile} & \text{Lemma 5.4.7.vi)}
\end{array}$$

This also contains a proof for $\Phi^{\smile};\Xi = \mathrm{X};\Psi^{\smile}$.

For the parameter component, we additionally have the following:

$$\begin{array}{rcl}
\mathsf{dom}\,\Omega_v &=& \mathsf{dom}\,(\Phi^{\smile};v_0;\Xi) \ = \ \mathsf{dom}\,(v_1;\mathrm{X};\Psi^{\smile}) \ = \ \mathsf{dom}\,(v_1;\mathrm{X})\\
\mathsf{ran}\,\Omega_v &=& \mathsf{ran}\,(\Phi^{\smile};v_0;\Xi) \ = \ \mathsf{ran}\,(\Phi^{\smile};v_0;\Phi;\mathrm{X};\Psi^{\smile}) \ = \ \mathsf{ran}\,(v_1;\mathrm{X};\Psi^{\smile}) \ = \ \mathsf{ran}\,(v_3;\Psi^{\smile})
\end{array}$$

The partitioning of $G_2$ into parameter part $v_2$ and non-parameter part $c_2$ can be copied from $G_3$:

$$
\begin{array}{rcl}
v_2 & := & \mathsf{ran}\,(v_0;\Xi) \;=\; \mathsf{ran}\,(v_0;\Phi;X;\Psi^{\smile}) \;=\; \mathsf{ran}\,(v_3;\Psi^{\smile}) \\
u_2 & := & \mathsf{ran}\,(u_0;\Xi) \;=\; \mathsf{ran}\,(u_0;\Phi;X;\Psi^{\smile}) \;=\; \mathsf{ran}\,(u_3;\Psi^{\smile}) \\
c_2 & := & \mathsf{ran}\,(c_3;\Psi^{\smile}) \\
v_3 & = & \mathsf{ran}\,(v_0;\Phi;X) \;=\; \mathsf{ran}\,(v_0;\Phi;X;\Psi^{\smile};\Psi) \;=\; \mathsf{ran}\,(v_0;\Xi;\Psi) \;=\; \mathsf{ran}\,(v_2;\Psi) \\
c_2 \sqcup v_2 & = & \mathsf{ran}\,(c_3;\Psi^{\smile}) \sqcup \mathsf{ran}\,(v_3;\Psi^{\smile}) \;=\; \mathsf{ran}\,((c_3 \sqcup v_3);\Psi^{\smile}) \;=\; \mathsf{ran}\,(\Psi^{\smile}) \;=\; \mathbb{I}
\end{array}
$$

With this, we can derive the two almost-injectivities:

$$
\begin{array}{rcl}
c_1 \sqcup \mathsf{dom}\,\Omega_v \sqcup X;\Psi^{\smile};\Omega_f^{\smile} & = & c_1 \sqcup \mathsf{dom}\,(v_1;X) \sqcup X;\Psi^{\smile};\Xi^{\smile};u_0;\Phi \\
& = & c_1 \sqcup \mathsf{dom}\,(v_1;X) \sqcup X;\Psi^{\smile};\Psi;X^{\smile};\Phi^{\smile};u_0;\Phi \\
& = & c_1 \sqcup \mathsf{dom}\,(v_1;X) \sqcup X;\Psi^{\smile};\Psi;X^{\smile};u_1 \\
& = & c_1 \sqcup \mathsf{dom}\,(v_1;X) \sqcup X;X^{\smile};u_1 \\
& = & X;X^{\smile} \\
c_2 \sqcup \mathsf{ran}\,\Omega_v \sqcup \Omega_u^{\smile};X;\Psi^{\smile} & = & c_2 \sqcup v_2 \sqcup \Xi^{\smile};u_0;\Phi;X;\Psi^{\smile} \\
& = & c_2 \sqcup v_2 \sqcup \Psi;X^{\smile};\Phi^{\smile};u_0;\Phi;X;\Psi^{\smile} \\
& = & c_2 \sqcup v_2 \sqcup \Psi;X^{\smile};u_1;X;\Psi^{\smile} \\
& = & c_2 \sqcup v_2 \sqcup \Psi;u_3;\Psi^{\smile} \\
& = & \mathbb{I} \\
& = & \Psi;\Psi^{\smile}
\end{array}
$$

The combination property:

$$
\begin{array}{rl}
\mathbb{I} = & \mathsf{ran}\,(\Phi;X) \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Phi)^{\sim};X) \sqcup (\mathsf{ran}\,X)^{\sim} \\
= & \mathsf{ran}\,((\mathsf{ran}\,\Phi)^{\sim};X) \sqcup u_3 \sqcup (\mathsf{ran}\,X)^{\sim} \sqcup v_3 \\
= & \quad \{\ \mathsf{ran}\,(v_2;\Psi) \sqsubseteq \mathsf{ran}\,(v_1;X)\ \} \\
& \mathsf{ran}\,((\mathsf{ran}\,\Phi)^{\sim};X) \sqcup u_3 \sqcup \mathsf{ran}\,((\mathsf{ran}\,\Xi)^{\sim};\Psi) \sqcup (X^{\smile};v_1;X \sqcap \mathsf{ran}\,(v_2;\Psi)) \\
= & \quad \{\ c_1;X \text{ and } \Psi \text{ univalent}\ \} \\
& X^{\smile};c_1;X \sqcup \Psi^{\smile};q_2;\Psi \sqcup (X^{\smile};v_1;X \sqcap \Psi^{\smile};v_2;\Psi)
\end{array}
$$

It remains to be shown that $\Xi$ is a host morphism, which means interface preservation and univalence of $u_0;\Xi$. Both follow in the same way as in the proof of Theorem 6.3.1, since that proof does not rely on almost-injectivity of X besides $u_1$. □

# Index


$\mapsto$, 32
$\nrightarrow$, 32
$\leftrightarrow$, 31, 67
$\rightarrow$, 32, 34, 46, 66, 86, 90
$\Rightarrow$, 86
$\diagdown$, 50, 87
$\backslash$, 85
$/$, 85
$:$, 31, 34, 66, 90
$\bullet$, 31, 32
$\vee$, 33
$\wedge$, 32
$\perp$, 33
$\top$, 33
$\leq$, 32
$\preccurlyeq$, 37
$\ll$, 44
$\bigvee_{\preccurlyeq,\mathcal{T}}$, 38
$\bigvee_{\ll,\mathcal{T}}$, 44
$\bigwedge_{\preccurlyeq,\mathcal{T}}$, 38
$\bigwedge_{\ll,\mathcal{T}}$, 44
$\bigvee$, 32
$\bigwedge$, 32
$;$, 66
$\mathbb{\perp}$, 68, 85
$\mathbb{T}$, 68, 70
$\sqcap$, 67
$\bigsqcap$, 69
$\sqcup$, 85
$\bigsqcup$, 85
$\sqsubseteq$, 67
$\times$, 34, 71, 90
$\_^{\smile}$, 67
$\_^{\neg}$, 46, 86
$\_^{\sim}$, 50, 87
$\_^{+}$, 102
$\_^{*}$, 102
$\_^{\boxast}$, 103
$\_^{\triangleright}$, 103
$\_^{\triangleleft}$, 103
$\overline{\_}$, 48

$\mathbb{1}$, 67, 71

abstract algebra, 73
$\Sigma$-$\mathsf{Alg}_{\mathbf{D}}$, 77
algebra, 37, 73
algebraic lattice, 34
allegory, 25, 67, 77, 84
allegory with sharp products, 72
almost-injective, 87
alphabet graph, 19
alternative commutativity, 147
alternative parameter commutativity, 184
antisymmetric, 68
application graph, 2
arity, 34
arrow, 66
atom, 34, 42
atomic lattice, 34

bijective, 68
Boolean lattice, 48, 58, 92
border, 54
border-discrete, 59

carrier, 37, 44
categorical product, 16, 26
category, 66
choice of direct products, 71
coherent, 117
combination property, 147
commute, 4
complement, 48
complete lattice, 33
completely distributive, 34, 41
completely downwards-distributive, 33
completely upwards-distributive, 33
component-free, 2
composition, 5, 66
conflict-free, 14, 107
congruence, 79
connected, 60, 95
constant symbol, 34
constructed gluing, 105
constructed pullout, 137
constructed tabulation, 101
context, 131
converse, 67

coregular part, 63
cospan, 98
crisp, 84

$\mathbb{D}_{\mathcal{A}}$, 96
$\mathbb{D}_{\mathcal{L}}$, 59
dangling condition, 12, 107, 142
data refinement, 73
declaration, 31
Dedekind category, 84–86
Dedekind formula, 67
Dedekind rule, 165
diagrammatic order, 5
difunctional, 103, 125
direct derivation, 9
direct gluing, 104
direct interface preservation, 135
direct product, 26, 70–72, 81–83
direct product of relations, 72
direct quotient, 80
direct result construction, 148
direct sum, 94
direct tabulation, 99
discrete, 7, 58, 95
discrete base, 59, 96
discrete part, 60
distributive allegory, 84
distributive lattice, 33, 85
dom, 69
domain of injectivity, 86
domain-discrete, 95
double-pullback approach, 142
double-pullback transition, 129
double-pullout approach, 140–143
double-pushout approach, 3, 7–12, 142
downwards-distributive, 33
duality, 33

edge replacement, 2
element, 65
empty carriers, 37
empty relation, 68
equivalence relation, 68, 80
essential part, 65
extended dangling condition, 142
extended identification condition, 140

$\mathcal{F}$, 34, 90
factorise, 5
family, 32
fibred approach, 130
function, 32
function symbol, 34
fuzzy, 84

glued tabulation, 135
gluing, 7, 104
gluing condition, 8–12, 107
gluing object, 7
gluing setup, 133
graph, 37
graph grammar, 1
graph homomorphism, 4
graph structure, 41, 92
graph transformation, 1
greatest element, 32
greatest lower bound, 32
$\Sigma$-$\mathsf{GS}_{\mathbf{D}}$, 92

Heyting algebra, 46
homogoneous relation, 68
homset, 66
host morphism, 8, 143
host object, 8
hyperedge replacement, 2

$\mathbb{I}$, 66
$\iota$, 94
ideal, 58
identification condition, 11, 107, 140
implication, 46
induced, 49
initial, 7, 67, 77
injdom, 86
injection, 94
injective, 68
injective part, 86
interface, 7, 22, 131
interface component, 132
interface part, 131
interface preservation, 132, 135
    direct, 135
interpretation, 36
ipa, 86
item, 4



join, 33, 85
join-distributivity, 85
join-irreducible, 34, 41

$\kappa$, 94
Kronecker product, 18

L-simulation, 73
labelled structure, 152
lattice, 32–34
lattice, distributive, 85
lazy double-pullback transition, 130
least element, 32
least upper bound, 32
left residual, 85
left-hand side, 2, 8
left-linear, 140
locally co-complete, 69, 85
loop edge, 4
lower bound, 32
lower semi-lattice, 32

Map, 68, 90, 99
mapping, 32, 68, 73, 77
matching, 2
meet, 32, 67
meet-subdistributivity, 67
modal rule, 67, 85, 165
$\mathsf{Mor}_{\_}[\_,\_]$, 66
morphism, 66

NLC graph rewriting, 19
node replacement, 2

object, 66
opfibration approach, 130
ordering, 68

$\pi$, 70, 71
parallel composition, 72
parameter, 22, 131
parameter component, 132
parameter part, 131
partial function, 68, 120
partial identity, 69, 70
partial morphism, 13, 120
partial unit, 69
Pfn, 68, 114, 120
PId, 69
point axiom, 84
product, 26, 70–72
production, 1
projection, 70
pseudo-complement, 46, 86
pseudo-complemented lattice, 46
PtoU, 45
pullback, 16, 99–102
pullback approach, 15–21
pullback complement, 116–120, 139
    maximal, 130
pullout, 134–138
    constructed, 137
pullout complement, 138–140
pushout, 4–7, 103–106
pushout complement, 8, 107–113
pushout object, 4

quotient, 79, 80

$\rho$, 70, 71
ran, 69
range-discrete, 95
reasonable gluing setup, 145
redex, 2
reduced gluing, 126
reflexive, 68
reflexive transitive closure, 102
regular, 49
*Rel*, 77
relation, 31, 67
relation algebra, 24–26, 92
relational homomorphism, 22, 74, 90
relational morphism closure, 88
relational morphism kernel, 93
relational structures, 74
relative pseudo-complement, 46, 86
relative semi-complement, 50, 87
replication, 15
residual, 85
restricted gluing, 127
restricting derivations, 12, 113
result embedding, 2
result graph, 2
rhs-morphism, 144
right residual, 85

right-hand side, 2, 8
$\mathsf{RMC}_{\Sigma,\mathbf{D}}$, 88
$\mathsf{RMK}_{\Sigma,\mathbf{D}}$, 93
rule, 1, 2, 56

$\mathcal{S}$, 34, 90
$\Sigma$-$\mathsf{Alg}_{\mathbf{D}}$, 77
$\Sigma$-algebra, 37, 73
$\Sigma$-algebra homomorphism, 77
$\Sigma$-compatible family of relations, 74
$\Sigma$-$\mathsf{GS}_{\mathbf{D}}$, 92
SAC, 39
SAK, 43
saturated symmetric quotient, 124
scalar, 84
semi-complement, 50, 87
semi-complemented lattice, 50
semi-injectivity, 147
separable part, 60
*Set*, 101
set comprehension, 31
sharp product, 70, 72
sharpness, 70
sigB1, 36, 38, 51
sigC1, 36, 39, 89
sigCLat, 35, 36
sigDHG, 35, 142
sigGL, 59
sigGraph, 35, 37, 89
sigGroup, 35, 36
sigL, 59
sigLat, 35
sigLoop, 36, 42
signature, 34
sigTwoSets, 36, 83
single-pushout approach, 3, 13–15
skeletal, 49
sloppy host construction, 111
solid, 60
    relation, 97
solid part, 61, 97
sort, 34
source, 66, 69
source tentacle, 4
span, 98
(spurious), 145
src, 34, 90
standard gluing morphism, 134
straight host construction, 111
strict Dedekind category, 87
strict unit, 71
strictly associative, 71
strongly connected component, 89
subalgebra closure, 39
subalgebra kernel, 43
subobject, 13, 78
subobject host construction, 111
surjective, 68
symmetric, 68
symmetric division, 124
symmetric quotient, 123

tabulation, 99–102
target, 66, 69
target tentacle, 4
tentacle, 4, 35
terminal, 16, 67
topological ordering, 89
total, 68
transformation, 2
transitive, 68
transitive closure, 102
trg, 34, 90

unary algebra, 41, 90
unary signature, 34, 90
unit, 69, 77
unit graph, 4, 16
unitary, 69
univalent, 68
univalent part, 86
universal characterisation, 5, 99
universal relation, 68, 70
upa, 86
upper bound, 32
upper semi-lattice, 33
upwards-distributive, 33

variable, 15, 22, 131
vertex replacement, 2, 15, 19

weak pullout, 146

Z, 31
zero law, 85
zero relation, 68