A Category Theoretical Approach to the Concurrent Semantics of Rewriting

Adhesive Categories and Related Concepts

Vom Fachbereich Ingenieurwissenschaften

der Universität Duisburg-Essen

zur Erlangung des akademischen Grades eines

Doktors rer. nat.

genehmigte Dissertation

von

Tobias Heindel

aus München

Referent: Prof. Dr. Barbara König

Korreferent: Dr. Paolo Baldan

Tag der mündlichen Prüfung: 01.12.2009

Abstract

This thesis studies formal semantics for a family of rewriting formalisms that have arisen as category theoretical abstractions of the so-called *algebraic* approaches to graph rewriting. The latter in turn generalize and combine features of term rewriting and Petri nets. Two salient features of (the abstract versions of) graph rewriting are a suitable class of categories which captures the structure of the objects of rewriting, and a notion of independence or concurrency of rewriting steps – as in the theory of Petri nets.

Category theoretical abstractions of graph rewriting such as double pushout rewriting encapsulate the complex details of the structures that are to be rewritten by considering them as objects of a suitable abstract category, for example an adhesive one. The main difficulty of the development of appropriate categorical frameworks is the identification of the essential properties of the category of graphs which allow to develop the theory of graph rewriting in an abstract framework. The motivations for such an endeavor are twofold: to arrive at a succint description of the fundamental principles of rewriting systems in general, and to apply well-established verification and analysis techniques of the theory of Petri nets (and also term rewriting systems) to a wide range of distributed and concurrent systems in which states have a "graph-like" structure.

The contributions of this thesis thus can be considered as two sides of the same coin: on the one side, concepts and results for Petri nets (and graph grammars) are generalized to an abstract category theoretical setting; on the other side, suitable classes of "graph-like" categories which capture the essential properties of the category of graphs are identified. Two central results are the following: first, (concatenable) processes are faithful partial order representations of equivalence classes of system runs which only differ w.r.t. the rescheduling of causally independent events; second, the unfolding of a system is established as the canonical partial order representation of all possible events (following the work of Winskel). Weakly ω-adhesive categories are introduced as the theoretical foundation for the corresponding formal theorems about processes and unfoldings.

The main result states that an unfolding procedure for systems which are given as single pushout grammars in weakly ω-adhesive categories exists and can be characetrised as a right adjoint functor from a category of grammars to the subcategory of occurrence grammars. This result specializes to and improves upon existing results concerning the coreflective semantics of the

unfolding of graph grammars and Petri nets (under an individual token interpretation). Moreover, the unfolding procedure is in principle usable as the starting point for static analysis techniques such as McMillan's finite complete prefix method. Finally, the adequacy of weakly ω-adhesive categories as a categorical framework is argued for by providing a comparison with the notion of topos, which is a standard abstraction of the categories of sets (and graphs).

Acknowledgments

First, thanks to my supervisor, Barbara König, who gave me the opportunity to work on a challenging topic and has guided me through a surprisingly tangled research area. Thanks to my second referee, Paolo Baldan, who was so kind to take this obligation upon himself. Thanks to my coauthors, Paolo Baldan, Filippo Bonchi, Andrea Corradini, Javier Esparza, Fabio Gadducci, Frank Hermann, Vitali Kozioura and Paweł Sobociński, who opened my eyes for many aspects of my work that I might have never noticed without them. Thanks to Filippo Bonchi, Vincenzo Ciancia, Barbara König, Johannes Riedl and Heiko Strathmann for their constructive criticism on drafts of this thesis. Thanks to Paweł Sobociński for being my companion on the way to the theoretical foundations of this thesis.

Thanks to Andrea Corradini and Fabio Gadducci for their invitations to spend several months in Pisa, which were more than fruitful periods of research. Thanks to Filippo Bonchi for many memorable evenings and journeys in and around Pisa. Finally, thanks to my parents for their steady support and their encouragements during all these years.

Contents

Introduction

1

> If someone were to analyse current notions and fashionable catchwords, [s]he would find 'systems' high on the list. The concept has pervaded all fields of science and penetrated into popular thinking, jargon and mass media.
>
> Ludwig von Bertalanffy [von Bertalanffy, 1973, page 1]

1.1 The 'systems' under investigation

The subject matter of this thesis is a certain class of formal system specifications, namely so-called *adhesive*[1] *rewriting systems* (ARS). In other words, the object of investigation is a family of mathematical objects which can be used to *describe* systems. Hence, before delving into the technical details of ARSs, some general considerations about the nature of systems might serve as an introduction – after all, adhesive rewriting systems are designed to specify *systems*, i.e. entities which one could try to characterize as "complexes of elements standing in interaction"[von Bertalanffy, 1973, page 32].

> In dealing with complexes of 'elements', three different kinds of distinctions may be made – i.e., (1) according to their *number*; (2) according to their *species*; (3) according to the *relations* of elements. [von Bertalanffy, 1973, page 53]

Take the Internet as an example of a system; it comprises a collection of computers connected by a number of network links, and the computers interact by sending and receiving messages. At a given time, there are only a *finite number* of computers, links and data. However, the Internet evolves over time: new computers are added, old links break down, new links are established. Hence there is no fixed maximal size it may have, neither is there

[1] As a note to the specialist, the adjective *adhesive* is taken from *adhesive categories*, which have been introduced in [Lack and Sobociński, 2004]. Rewriting in adhesive categories is a generalization of double pushout rewriting of graphs [Ehrig et al., 1973] and has been generalized to adhesive high-level replacement systems [Ehrig et al., 2004b]; however, some of the underlying and related ideas can already be found in [Kennaway, 1990] and [Monserrat et al., 1997]. In this thesis, adhesive rewriting systems are understood to encompass all these and similar approaches to rewriting; in particular ARS does not only refer to double pushout rewriting systems in (properly) adhesive categories.

a foreseeable point in time when it will cease to exist or "halt". In this sense one may conceive of the Internet as a possibly infinite, distributed, dynamic system.

An exact taxonomy of the different *species* of the elements of the Internet would probably be a research project in its own right. Hence the exact nature of the elements the Internet might consist of is left open; however, note that – depending on the level of abstraction and the context – different options are viable. Here it only matters that computers, links, data, messages etc. do form elements of the Internet, and even a layman has rudimentary knowledge about what these words refer to.

Finally, the *relations* between the elements are manifold: computers are connected with each other via links, data are stored on computers, messages are exchanged between computers, new computers are added, links break down, and so forth. Looking at this list of possible relations it becomes apparent why the Internet is the prime example of a *distributed* system. However, these relations my also change over time, e.g. computers may cease to be connected when the links between them break. This latter fact witnesses that the Internet is also *dynamic* in nature.

Examples of other systems abound as an Internet search for the word '*system*' reveals. The development of a theory that captures any conceivable system is – or was – the goal of Bertalanffy's General System Theory. In contrast, the class of systems which can be described faithfully using the *formal* methods of computer science might appear small. However, even if not all aspects of a given system can be accounted for using a specific formalism, it might still be possible that at least some aspects can be represented in a formal system. In this way, a complex (real world) system can give rise to (the formal specification of) a new and simplified "abstract" system.

The systems that will be studied in this thesis, namely *adhesive rewriting system*, are of the latter kind, i.e. abstract systems, and – pushing abstraction one step further – they might not even be the result of any modelling endeavor. In other words, an ARS is treated as a mathematical object. However, this does not imply that the study of these mathematical objects cannot be useful for purposes of the specification, analysis and verification of (software) systems. As a typical example, it might help to develop a general method for a whole class of "complexes of elements standing in interaction"[von Bertalanffy, 1973]. The first two concrete examples of such classes are graph transformation systems and Petri nets.

1.2 First examples of adhesive rewriting systems

From a software engineering perspective, the use of adhesive rewriting systems as a specification tool goes hand in hand with implicit assumptions about systems. First, at any time, the described system must have a discernible, stable structure, which is referred to as a *system state*. In the case of graph transformation, states are graphs, and each state of a Petri net is given by the number of tokens in each place, i.e. a multiset over the set of its places.

Second, all changes that take place during any finite period of time, i.e. any *transition* from one state to another one, can be modeled by a number of (possibly concurrent) executions or applications of a fixed set of *rules*; the latter describe how the elements of a system can interact, evolve and affect the structure. A graph transformation rule is illustrated in Figure 2 where '$\supseteq$' and '$\subseteq$' indicate that the middle graph is a subgraph of the left and right one, respectively.

Ideally, the rules describe which *local* modifications of states are possible, and they can be thought of as explanations for the structural changes in the system, or as laws that all possible evolutions of the system obey. Often, rules are used to specify the possible interactions between components of system states. This will be illustrated using the following graph transformation example in the context of the Internet.

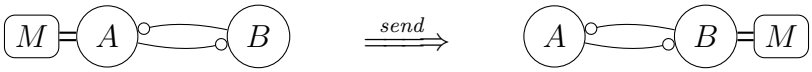
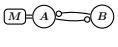
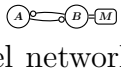


Figure 1: Modeling a sub-system of the Internet: *send*ing M from A to B

Figure 1 illustrates how the exchange of a message M between two computers A and B might be modeled – on a rather high level of abstraction. The figure displays two graph representations of system states, namely the two structures  and  . Round nodes model network nodes, the edges of the form —∘ model network links; further messages correspond to boxes and the fact that a message is located at a network node is modelled by a double line. Moreover the figure indicates that a transition from the first to the second state is possible, and that such a transition can be "explained" by means of the *send*-rule, which is depicted in Figure 2.

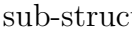
A transition of type *send* leads from a state X to a state Y whenever X contains a sub-structure of the form  , i.e. (a copy of) the *left hand side* of the *send*-rule *occurs* as a sub-structure of X. The transition from X

Figure 2: The *send*-rule, which models the forwarding of messages

to Y amounts to replacing such an occurrence of the the left hand side of the rule by the structure ○–○–□ – the *right hand side* of the *send*-rule. However, during this replacement process, the *interface* or *gluing object* □ ○–○ of (the occurrence of) the left hand side of the *send*-rule remains unaffected.

Speaking in terms of chemical reactions, the gluing object behaves like a catalyst of the transition, and the replacement process roughly fits the framework of the chemical abstract machine [Berry and Boudol, 1992]. In the present example, the *send*-rule expresses that only the location of the message changes while the message itself, the communication partners and the link between them are left unchanged.

Summarizing, the replacement process destroys what is present in the left hand side but not in the interface, and it adjoins the new portion of the right hand side, i.e. everything that is not already present in the gluing object is glued to the remainder of the (occurrence of the) left hand side. In the example, only the connection from M to A is destroyed, and a new one from M to B is created. Hence, as a rough description, a *rewriting step* consists of a deletion

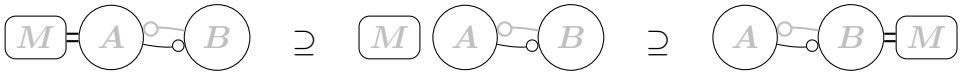


Figure 3: An explanation of *send*ing M from A to B with two phases

and a construction phase, and it temporarily passes through an "unstable" intermediate state as illustrated in Figure 3, where the gray parts give the context in which the rule is applied.

However, the *send*-rule is also applicable to the new state ○–○–□, and therefore the message M might oscillate back and forth between A and B. Hence, so far, this message sending system has two states and exactly one infinite run or "computation". When adding a second message to this message passing system, say □–○–○–□, there are four different states and the two messages can travel independently of each other, or even simultaneously, i.e. *concurrently*.

Concurrency of transitions and events. The phenomenon of concurrency manifests itself in a paradigmatic way in the theory of Petri nets. A Petri net consists of a set of places and a set of transitions, and each transition may consume and produce tokens which reside in the adjacent places of the transition. Places are depicted as empty circles ◯, and transitions as boxes ■. Each transition is connected to some of the places via arcs ⟶▷, e.g. ◯▷■▷◯. A token is represented as a disc ● and may reside only on places, say ◉▷■▷◯. The latter state or *marking* is described more succinctly by the pair $(1, 0)$, which indicates that there resides one token on the first place and none on the second.

In the latter state $(1, 0)$, the transition is *enabled* because the source place of each incoming arc contains a token. Hence, the transition can *fire* by consuming the token(s) along the incoming arc(s) and producing a new token at the target place of each outgoing arc; the resulting state is $(0, 1)$, which is depicted as ◯▷■▷◉.

Now, consider the two Petri net models of the message passing system which are displayed in Figure 4. In each of the two nets $\mathcal{N}_1$ and $\mathcal{N}_2$, the transitions

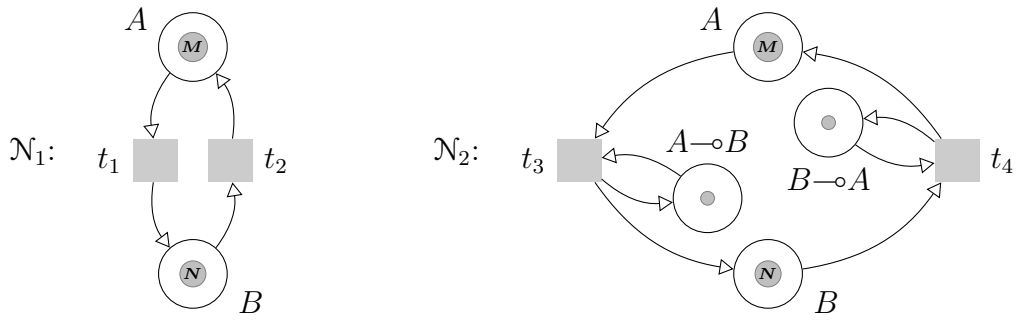


Figure 4: Two Petri net models of the message passing system

correspond to the several ways in which the *send*-rule can be applied, namely from A to B or in the opposite direction.

The net $\mathcal{N}_2$ is more exact insofar as it represents the links between A and B which the *send*-rule depends on – this is done in an ad hoc manner by adding two extra places with annotation $A \multimap B$ and $B \multimap A$, respectively. However the net $\mathcal{N}_2$ deletes and re-establishes these links whenever one of the transitions t_3 or t_4 is fired. As a consequence, the net $\mathcal{N}_1$ is more exact w.r.t. to concurrent execution of transitions in the following sense.

Both nets with the markings displayed in Figure 4 can fire their transitions simultaneously: $(M, N)[t_1 t_2\rangle(N, M)$ in $\mathcal{N}_1$ and $(M, 1, N, 1)[t_3 t_4\rangle(N, 1, M, 1)$

in $\mathcal{N}_2$. The difference is seen as follows: whereas $\mathcal{N}_1$ can execute t_1 once, followed by two simultaneous firings of t_2, the net $\mathcal{N}_2$ cannot simulate such an action sequence. In other words the firings $(M, N)[t_1\rangle(0, MN)[2t_2\rangle(MN, 0)$ in $\mathcal{N}_1$ cannot be matched in $\mathcal{N}_2$, because after executing $(M, 1, N, 1)[t_3\rangle(0, 1, MN, 1)$ the place $B \multimap A$ contains only one token. In this situation, the best approximation would be two *sequential* executions – either transmitting first M followed by N or the other way around, i.e. $(0, 1, MN, 1)[t_4\rangle(M, 1, N, 1)[t_4\rangle(MN, 1, 0, 1)$ or $(0, 1, MN, 1)[t_4\rangle(N, 1, M, 1)[t_4\rangle(MN, 1, 0, 1)$. This means there is a choice of two different *interleavings* of actions in $\mathcal{N}_2$.

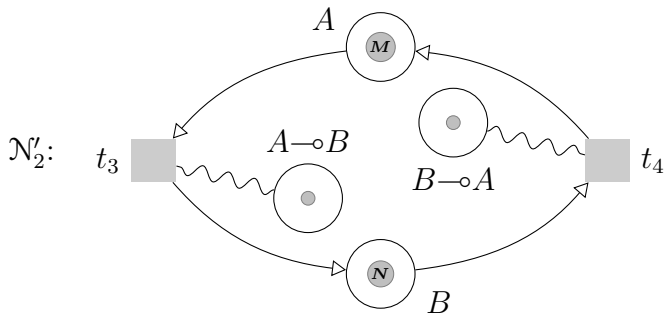


Figure 5: An improved Petri net model of the message passing system

This lack of concurrency in $\mathcal{N}_2$ can be mended by introducing *read arcs*, which are the Petri net counterpart of gluing objects; they only check for the presence of tokens in a place without consuming them. Tokens in places that are adjacent to such read arcs potentially play the role of "catalysts" of the transitions. The resulting variant $\mathcal{N}_2'$ with read arcs is shown in Figure 5, where read arcs are drawn as curly lines. In this way – with the additional information that tokens in the places $A \multimap B$ and $B \multimap A$ correspond to edges from A to B, and B to A, respectively – the net $\mathcal{N}_2'$ captures the relevant information about the message passing system.[2]

In the reverse direction, Petri nets can be seen as special cases of adhesive rewriting systems. However, the message sending system of Figure 1 is a better example of an ARS insofar as it illustrates that states may contain non-trivial connectivity information whereas Petri net markings (without annotations as above) are discrete. The topic of structured, "graph-like" states is addressed

[2]A general method which can be used to represent graph transformation systems using so-called Petri graphs, which are Petri nets with a suitable graph annotation, is developed in [Baldan et al., 2008c].

in Section 1.3.2.

1.3 Two guiding ideas of the investigation

The introductory example of a message passing system already touched upon some central notions, namely *states*, *transitions*, and *concurrency*; further it illustrated the principle that transitions are in one-to-one correspondence to *rule applications*. However the following two guiding ideas are still missing.

The first idea is the concept of the *unfolding* of a given system into a complete, explicit system description which is obtained by recording all possible transitions with information concerning their history and causal dependencies. In what sense unfoldings completely describe systems is sketched in Section 1.3.1, based on a Petri net model for the above message passing system as example.

The second guiding idea is the aim for a uniform, abstract construction principle for such unfoldings which only depends on the fact that system states can be modelled as structures which – on a suitable level of abstraction – relate to each other in the same way as graphs relate to each other. In technical terms, system states should be objects of a category which shares "enough" properties with the category of graphs and graph morphisms. Section 1.3.2 provides an informal presentation of such "graph-like" structures.

1.3.1 Unfoldings as explicit system descriptions

The unfolding of a system represents the collection of all *runs* of a systems starting from a given initial state, say X_0. Here, a run from X_0 is a finite, possibly empty sequence of transitions $X_0 =_{T_1}\Rightarrow X_1 =_{T_2}\Rightarrow X_2 \cdots =_{T_n}\Rightarrow X_n$ in the system; repetitions of states are allowed, which means that $X_i = X_j$ does not imply $i = j$. However, of all the runs, the unfolding only records the *local* (typically small) changes of each transition. The reason is that each transition is a rule application and thus changes only occur in the "neighbourhood" of the occurrence of the left hand side of the rule.

The crucial consequence is that no spurious causal dependencies between the transitions of runs are introduced in unfoldings. This is in contrast to standard semantics known from the area of process calculi, where all possible transition sequences are merely "assembled" in a labelled transitions system (LTS): for each transition sequence, the LTS also contains all its *interleavings*, which are obtained by repeatedly switching pairs of causally independent transitions. Indeed, the unfolding based system representation avoids the

resulting state space explosion problem of highly concurrent systems. This is achieved by exploiting the fact that the causal relation between transtions is in general a partial order.

The partial order of causality is closely related to the principle of local change: the whole information of a transition $X =T\Rightarrow Y$ is given by the location in X at which a rule has been applied and that part of Y which corresponds to the right hand side of the applied rule. This information amounts to a copy of the applied rule, which is called a rule *occurrence*; its left and right hand side are suitably embedded into X and Y.

Now, to construct the unfolding from the start state X_0, it suffices to consecutively extend the start state X_0 by adjoining (copies of) the right hand sides of the applied rules of each run $X_0 =T_1\Rightarrow X_1 =T_2\Rightarrow X_2 \cdots =T_n\Rightarrow X_n$, and to record the locations at which occurrences of left hand sides have been identified and (the new portions of) right hand sides have been added.

For the special case of Petri nets, the situation is as follows: given a transition in a Petri net (without read arcs), the "left hand side" is its pre-set, which is the (multi-)set of places with arcs *into* the transition, and similarly, the "right hand side" is the post-set, i.e. the (multi-)set of places with arcs *from* the transition; finally the "gluing object" is always the empty marking.[3] Hence, for each transition occurrence in a net, only the post-set needs to be added at the appropriate location.

More precisely, the unfolding procedure for Petri nets can be sketched as follows. Given a net with some (initial) marking, the construction starts with a "copy" of the initial marking and no transition occurrences. After this initialization, it proceeds by repeatedly adding (copies of) transition occurrences and (copies of) their post-sets. As an example, consider the (system described by the) Petri net $\mathcal{N}_1$ from Figure 4. Its unfolding[4] is sketched in Figure 6. The gray numbers indicate an order in which occurrences of transitions with the corresponding places in the post-set (could) have been added to the unfolding; as a crucial fact, any other order would yield the same result.

To unfold the net $\mathcal{N}_1$ into $\mathcal{U}$ proceed as follows. Start with a set of places which contains for each token of the initial marking a copy of the corresponding place; in the example this is the marking consisting of M at A and N at B, i.e.

[3] In a Petri net with read arcs, the "gluing object" of a transition consists of the sum of all places that are read, and these places have to be added to the pre- and post-set to obtain the "left and right hand sides", respectively.

[4] Unfoldings of Petri nets have been introduced in [Nielsen et al., 1981]; nevertheless, the informal descriptions of unfoldings are inspired by [McMillan, 1995].

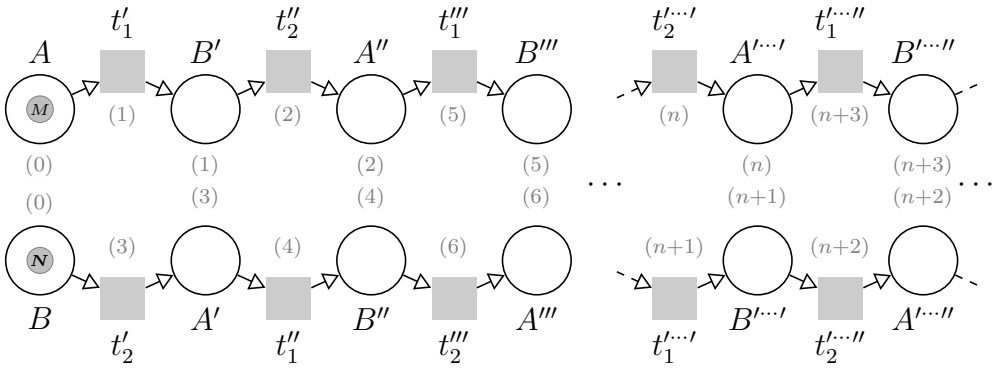


Figure 6: The unfolding $\mathcal{U}$ of the the net $\mathcal{N}_1$

the starting point are the two places labelled with (0) without any transition occurrence. Then repeatedly apply the following two steps.

1) Choose a reachable marking of the unfolding $\mathcal{U}_{(n)}$ constructed so far, i.e. fire a possibly empty sequence of transitions in the unfolding $\mathcal{U}_{(n)}$. Execute the corresponding sequence of transitions in the original net $\mathcal{N}_1$ – this is a *run* of the original net which is already represented in $\mathcal{U}_{(n)}$.

   As $\mathcal{N}_1$ cannot reach dead-lock situations, there is always a transition t_i in $\mathcal{N}_1$ that can be executed next. Further, w.l.o.g., the chosen sequence of transitions in $\mathcal{N}_1$ is such that the transition t_i in $\mathcal{N}_1$ does not yet correspond to any transition in $\mathcal{U}_{(n)}$ (otherwise fire additional transitions in $\mathcal{U}_{(n)}$ until one of the "rightmost" places contains a token). More concretely, at least one token labelled with M or N is in a "rightmost" place in $\mathcal{U}_{(n)}$; moreover, choosing one of these tokens, there is exactly one corresponding transition t_i in $\mathcal{N}_1$ that is enabled.

2) Adjoin a copy of t_i to $\mathcal{U}_{(n)}$: e.g. this can be a transition occurrence $t_1^{\prime\cdots\prime}$ of t_1 which has the place $A^{\prime\cdots\prime}$ in $\mathcal{U}_{(n)}$ as its pre-set; its post-set is a new copy of B, say $B^{\prime\cdots\prime\prime}$, and similarly for $t_2^{\prime\cdots\prime}$. Hence, the next unfolding $\mathcal{U}_{(n+1)}$ has exactly one new transition occurrence and one new place (though in general, the post-set of a transition might be larger or empty).

Finally, the full unfolding $\mathcal{U}$ of the net $\mathcal{N}_1$ arises as the union of all the partial unfoldings $\mathcal{U}_{(n)}$, and thus is potentially infinite. By construction, if all choices are made fairly, each run in the original net $\mathcal{N}_1$ is represented as some run in the unfolding $\mathcal{U}$. This is known as the (full) completeness of unfoldings.

Unfoldings have been introduced for Petri nets in; they have been generalized to graph transformation in [Baldan, 2000]. In this thesis, the unfolding construction is generalized to the abstract framework of adhesive rewriting systems. However, the main contribution concerning unfoldings is due to an improvement of the grammar morphisms of [Baldan, 2000] which allows to characterize the unfolding procedure for adhesive rewriting systems as a (co-)free construction in the style of [Nielsen et al., 1981]. This result can be considered application relevant for the following two reasons.

First, the (co-)freeness of the unfolding construction ensures that it is sound to unfold a composed system (cf. [Winskel, 1985]) by composing the unfoldings of the components of the system. This *divide and conquer* approach is particularly useful in the case of distributed systems, as the unfoldings of each system component can be computed locally (cf. [Baldan et al., 2006b]).

Second, since the (co-)freeness of the unfolding construction is established in the abstract framework of adhesive rewriting systems, it specializes to rewriting of various "graph-like" structures. Hence, system engineers that plan to use unfoldings, e.g. for verification purposes, can choose appropriate models for system states from a large but uniform class of structures and are not forced to encode information in some particular fixed formalism (cf. [Ehrig et al., 2006]).

1.3.2 "Graph-like" states as objects of a (weakly) adhesive category

The second guiding idea is to clarify what it means (in the context of the theory of unfoldings of rewriting systems) that (a given class of) structures can be considered as "sufficiently graph-like". As a preliminary answer, the presupposition is that such structures should have (at least) the following two characteristic properties: first, there exist suitable "clipping" and "gluing" operations which are compatible with each other such that rewriting steps can be described in terms of a deletion and construction phase as illustrated in Figure 3; second, it is possible to construct the "union" of a growing chain of structures to obtain a single structure such that any chain of unfoldings can be coalesced into a single (full) unfolding.

The words "clipping" and "gluing" can be illustrated with a metaphor which is based on geometric intuition and assumes that any "graph-like" structure can be faithfully represented by a drawing on a sheet of some ethereal "ideal fabric" as shown on the left in Figure 7.

Moreover, to extend this metaphor to *morphisms* from one structure to another one, for example to account for operations that map nodes and edges of one graph to nodes and edges of another one in a structure preserving

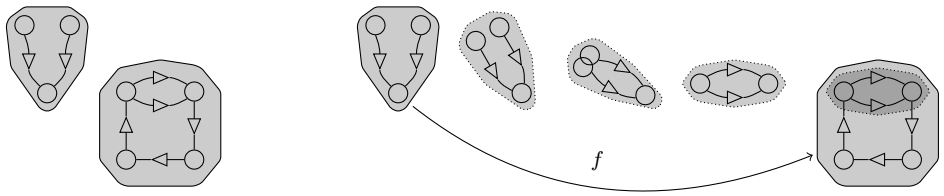


Figure 7: Illustration of two graphs and a graph morphism

way, it is necessary that this "ideal fabric" can stretch and shrink indefinitely and that even intersections of the fabric with itself are possible. Given such an "ideal fabric", a morphism between graphs can be realized as a morphism between the sheets on which the graphs are drawn; the illustration in Figure 7 shows how such a (realization of a) morphism continuously deforms one sheet of fabric to (part of) another one.

Speaking in terms of this metaphor, the fundamental idea of a category theoretical approach is to abstract away from the concrete "graph-like" structures and to focus on the properties of the "ideal fabric" and how its morphisms relate to each other. The crucial task is to express the "clipping" and "gluing" operations of structures and their compatibility on this abstract level.

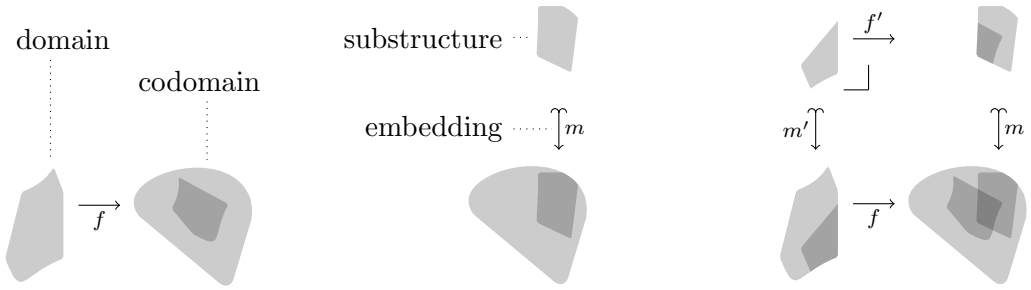


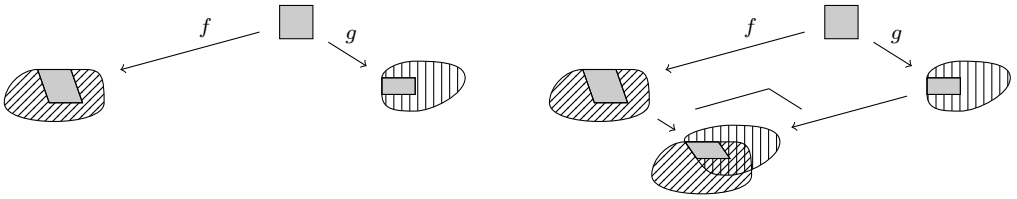
Figure 8: Illustration of a morphism, an embedding, and clipping $\downarrow\overrightarrow{\square}\downarrow$

Appealing to geometric intuition, an arbitrary morphism f between structures is depicted as an arrow as shown on the left in Figure 8; its origin and destination are called the *domain* and *codomain* of f, respectively. For example f could be a function between two sets, or an algebra homomorphism, or any continuous map between spaces. To keep the analogy with these concrete examples, the *image of* f in its codomain is marked as a darker area.

Next, "clipping" of structures is performed w.r.t. to a class of *embeddings*,

which consists of special "inclusion"-morphisms that embed one structure into another one. Typical examples of embeddings are inclusion of sets, injective functions or homomorphisms, or inclusions of open subspaces. Embeddings are depicted as shown in the middle of Figure 8. Given a morphism f and an embedding m of a substructure of the codomain of f, "clipping" is performed by pulling the embedding back against the morphism. As illustrated on the right in Figure 8, the result is the *pre-image m' (under f)* of the embedding m and the *(codomain) restriction of f' (to m)*. In the case of sets, given a function $f\colon A \to B$ and an injective function $m\colon M \rightarrowtail B$, the pre-image $m'\colon M' \rightarrowtail A$ is the usual one, i.e. the inclusion of the set $M' = \{a \in A \mid f(a) \in m(M)\} \subseteq A$ where $m(M) = \{m(x) \mid x \in M\}$; moreover $f'\colon M' \to M$ maps each $x \in M'$ to the unique $y \in M$ which satisfies the equation $m(y) = f(x)$.

Further, "gluing" of structures can be performed for any *span* of morphisms, i.e. for any a pair of morphisms f and g with a common domain. An example



(a) Span of morphisms (b) The gluing of f and g

Figure 9: Gluing of structures $\downarrow\!\!\overleftrightarrow{\downarrow}$

of such a span is illustrated in Figure 9(a). In this situation, the morphisms f and g deform a square in two different ways and embed it into the respective codomains. Now, the gluing of f and g is performed by identifying the two images of the square and adjoining the remainder of the codomains of f and g as illustrated in Figure 9(b) (this construction might involve certain local "smooth" changes near the border of the (images of the) deformed square). In the case of sets, given functions $f\colon A \to B$ and $g\colon A \to C$, the gluing of f and g is the fibred sum $B +_A C$, i.e. the quotient $(B + C)/_{\sim_f^g}$ of the disjoint union $B + C = \{0\} \times B \cup \{1\} \times C$ where $\sim_f^g$ is the least equivalence relation containing $\big\{\big\langle\langle 0, b\rangle, \langle 1, c\rangle\big\rangle \mid \exists a \in A.\, f(a) = b \,\&\, g(a) = c\big\rangle\big\}$.

Finally, the central property of the "abstract fabric" is that gluing and and clipping are compatible with each other. To explore what this is supposed to mean, consider two embeddings m_1 and m_2 of substructures into the respective codomains of f and g which have the same pre-image in the common domain of

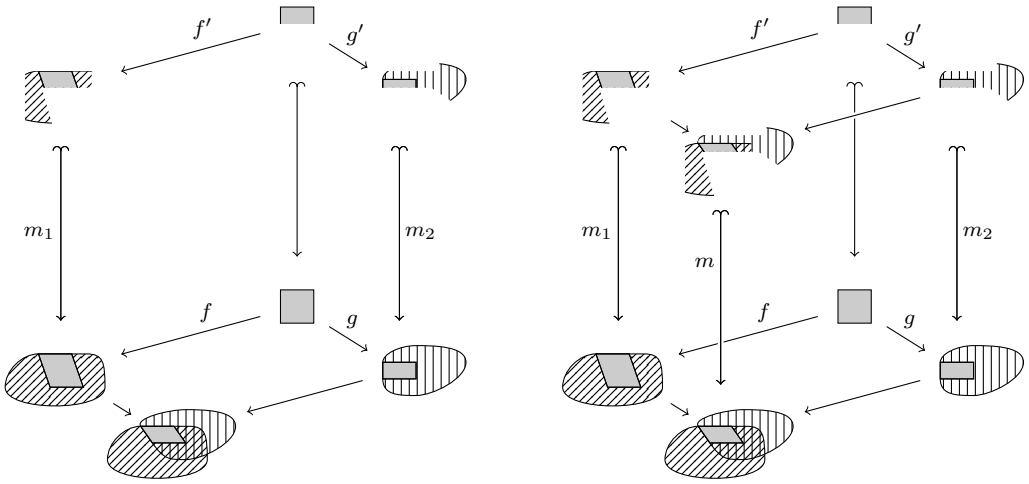


Figure 10: Compatibility of gluing and clipping

f and g, namely the upper half of the square (see the left diagram in Figure 10). Now, for "graph-like" structures, the following equivalence is postulated for any embedding m into the gluing of f and g (cf. the right hand diagram in Figure 10): the original embeddings m_1 and m_2 arise as pre-images of m (w.r.t. gluing of f and g) if and only if the embedding m arises from the gluing of the restricted morphisms f' and g'. Moreover, for every embedding m into the gluing of f and g, there exist suitable restrictions f' and g' such that m arises from the gluing of f' and g'.

The formal counterparts of the above metaphorical terms are as follows: structures are objects of a category $\mathbb{C}$ with an admissible class of monomorphisms $\mathscr{M}$, the latter being the formal counterpart of embeddings; moreover "gluing" is nothing else but taking pushouts, and "clipping" corresponds to the construction of pullbacks along $\mathscr{M}$-morphisms. The central is fact that gluing and clipping are compatible with each other as described in the previous paragraph, i.e. that pushouts are compatible with pullbacks. This compatibility condition can be shown to be equivalent to the requirement that pushouts in the category $\mathbb{C}$ induce pushouts in the associated, larger category of *partial maps*[5], which is a suitable abstraction of the category of sets and partial functions (cf. [Kennaway, 1990, Robinson and Rosolini, 1988]).

[5]In the category of partial maps, morphisms originate from a substructure of their domain, the so-called *domain of definition*, which is embedded using one of the morphisms of the admissible class $\mathscr{M}$.

Now, in the context of double pushout rewriting [Ehrig et al., 1973], one of the implications of the theoretical developments of this thesis is that compatibility of pushouts with pullbacks suffices to derive the major part of the results that have been established for adhesive categories [Lack and Sobociński, 2005] and related formalisms [Ehrig et al., 2004b, Ehrig and Prange, 2006]. Further, the proposal of weakly $\mathscr{M}\omega$-adhesive categories (Definition 5.14) as a framework for the unfolding of arbitrary adhesive rewriting systems is mainly based on this property of pushouts (and also transfinite compositions of embeddings). In fact, weakly ($\mathscr{M}\omega$-)adhesive categories can be considered to be a compromise between the theoretical elegance of adhesive categories and the wider applicability of the weak adhesive HLR categories of [Ehrig et al., 2004b, Ehrig and Prange, 2006].

However, the definition of weakly ($\mathscr{M}\omega$-)adhesive categories is also closely related with the results of [Johnstone et al., 2007] concerning the relation between (quasi-)adhesive categories and (quasi-)topoi [Johnstone, 2002]. The latter are among the most well-known classes of *abstract*[6] categories in which objects and morphisms relate to each other as one would expect from concrete categories like the category of sets and functions or the category of graphs and graph morphisms. The main result in this context is that any (quasi-)topos with countable coproducts is a (weakly) $\mathscr{M}\omega$-adhesive category. This fact alone demonstrates the generality of the proposed framework as well as its consistency with an established abstraction of "set-like" categories.

1.4 Thesis synopsis

This thesis revisits results of selected publications of the author. The main text is divided into two parts: the first one is oriented towards applications and presents results that provide a uniform perspective on a wide range of models for concurrent computational systems; the second one discusses suitable category theoretical frameworks for concurrent computational systems with "graph-like" states and provides the theoretical foundation for the results of the first part using standard notions of (basic) category theory (which are summarized in Appendix A). The logical dependencies of the two parts are sketched in Figure 11; the inner structure of the two parts and their related

[6] An abstract category is a category which – in contrast to a concrete category – does *not* come equipped with a suitable functor into the category of sets. In particular, deviating from Bourbaki-an lore, objects are *not* assumed to be sets with structure, and morphisms are *not* required to be functions that preserve the structure of the concrete objects.

work in the literature are illustrated in Figure 12 and Figure 13.

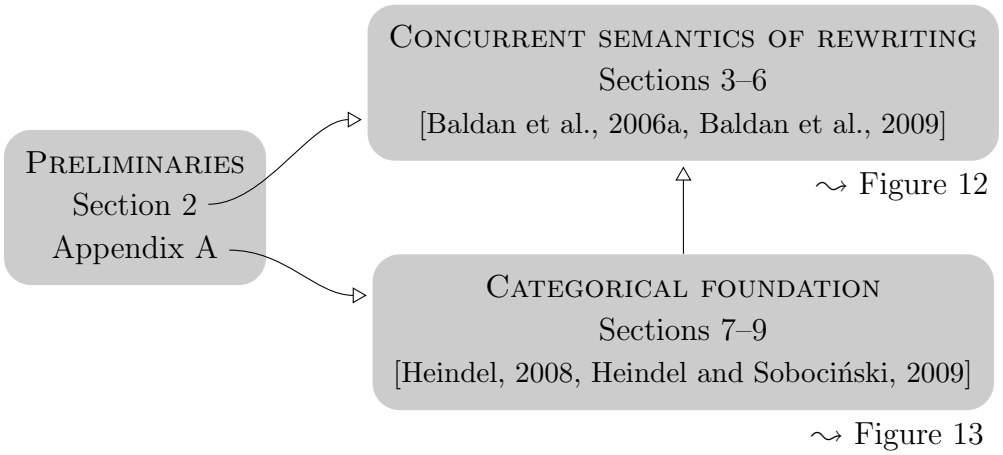


Figure 11: Logical dependence of the central parts of the thesis

The first part concerns the concurrent semantics of rewriting of objects in abstract categories. It studies processes as a true concurrency model of computation in the style of Goltz and Reisig, and describes unfoldings as canonical causal descriptions of event based systems following the ideas of Nielsen, Plotkin and Winskel. The corresponding publications of the author are [Baldan et al., 2006a] and [Baldan et al., 2009], respectively.

The second part provides the categorical foundation for the results in the first part. The two relevant publications on this theoretical background are [Heindel, 2008] and [Heindel and Sobociński, 2009]. The former introduces weakly adhesive categories as an alternative to the weak adhesive high-level replacement categories of Ehrig and Prange. The latter is a fundamental study of the main defining property of the adhesive categories of Lack and Sobociński.

Finally, the preliminary, basic notions of category theory are laid out in Appendix A and the central definitions and facts about the relevant rewriting formalisms, i.e. the single and double pushout approach, are presented in Section 2. As indicated by the arrows in Figure 11, the general ideas of the main results of the first part should be accessible even without detailed knowledge of the categorical foundations. Auxiliary results and the major part of the proofs are listed in Appendices B–E.

The above core publications of the author are accompanied by the following ones about related topics and applications (see Table 1 on page 30 for a list of the publications by the author). The interplay between inde-

pendence and parallel or synchronous execution of rewriting steps is studied in [Bonchi and Heindel, 2006] and [Bonchi et al., 2008] w.r.t. two different variants of double pushout rewriting, namely the double pushout approach with monic matches and its "interactive" extension with so-called *borrowed contexts*. Moreover, a generalization of the double pushout approach with the ability to clone substructures in individual rewriting steps is proposed in [Corradini et al., 2006]. Further, [Baldan et al., 2005] is a case study on the verification of red-black trees using graph transformation systems. A proposal for the formalization of the concept of secrecy in adhesive categories is discussed in [Heindel, 2009]. Finally, a study of the notion of irreducibility from lattice theory applied to objects in adhesive categories can be found in [Baldan et al., 2008a].

Structure of the parts and related work. The first part on the concurrent semantics of rewriting can be seen as a continuation of Paolo Baldan's research on the concurrent semantics of graph rewriting as summarized in his PhD thesis [Baldan, 2000]. Improved and generalized versions of his results are presented in the first four main sections of the present thesis. Each section starts with an analogy to one influential idea in the area of concurrency theory, namely Mazurkiewicz traces [Mazurkiewicz, 1986], partially ordered multisets (pomsets) [Pratt, 1982] (as an equivalent alternative to Mazurkiewicz traces), the unfolding of Petri nets in the algorithmic style of [McMillan, 1995], and the characterization of unfolding as a (co-)free construction [Nielsen et al., 1981], respectively. These four ideas circumscribe the general area within computer science in which the investigation takes place, and they are listed to the right in Figure 12 with the corresponding sections to the left.

The starting point is the formalization of concurrency in Section 3. The fundamental observation is the following: two (types of) actions might be independent of each other; in this case, the order in which the two independent actions are performed can be considered immaterial. In the theory of Mazurkiewicz traces, each action type corresponds to a letter of some alphabet, and this alphabet is endowed with an irreflexive, symmetric relation which models the independence of the different actions. Sequences of letters which can be obtained from each other by repeatedly switching pairs of adjacent, unrelated letters are considered equivalent; each equivalence class is a Mazurkiewicz trace and models a computation path up to concurrency.

In double pushout rewriting, sequences of actions or computation paths are modeled as sequences of rewriting steps, which are also referred to as

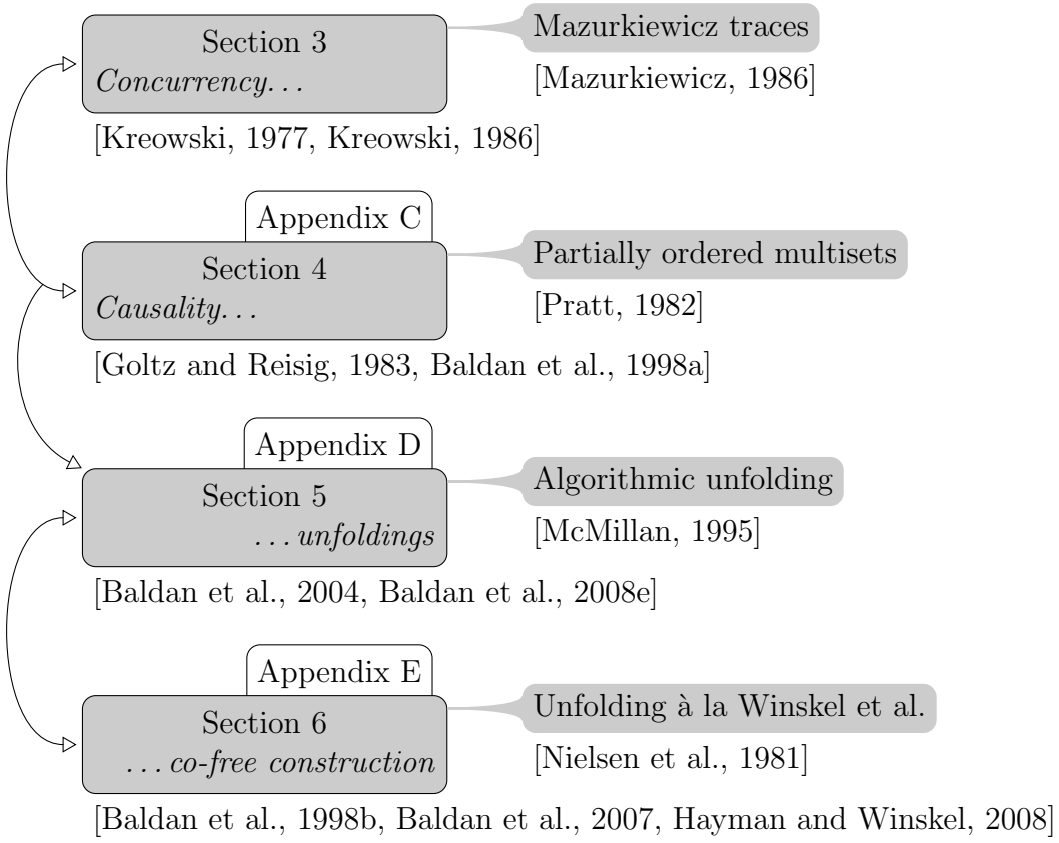


Figure 12: Structure of the first part *Concurrent Semantics of Rewriting* and some related work on graph transformation and Petri nets

derivations. The rewriting formalism is rule based and acts on objects of suitable abstract categories. Moreover, there is a standard notion of independence of pairs of consecutive rewriting steps. Hence, again, a computation path up to concurrency will be modeled as an equivalence class of such rewriting sequences where two sequences are considered equivalent if they can be obtained from each other by repeated switching of pairs of independent rewriting steps. However, already in the case of graph rewriting, the switching of pairs of rewriting steps is non-trivial.

The purpose of Section 3 is to introduce the counterpart of Mazurkiewicz traces for sequences of double pushout rewriting on "graph-like" structures;

technically, rewriting acts on objects of weakly adhesive categories, a generalization of the adhesive categories of [Lack and Sobociński, 2004] inspired by [Ehrig and Prange, 2006]. In the area of graph rewriting, the idea of switching of derivation steps is already used implicitly in [Kreowski, 1986].

The latter work also considers the parallel execution of concurrent rewriting steps, whereas the present thesis only considers "plain" switching using the constructions of the proof of the so-called Sequential Commutativity Theorem (see [Habel et al., 2001]). One might object that in this way, the theory of canonical "maximally parallel" normal forms of sequences of rewriting steps of [Baldan, 2000] is lost. However, a much less complex alternative to these normal forms will be presented in Section 4. Another reason against parallel steps is that it is not clear how parallel executions can be performed in weakly adhesive categories, in general.[7]

Section 4 presents an alternative but equivalent view on concurrency. The central idea is that two actions are independent if and only if they are not causally related. Further, equivalence classes of *totally ordered* sequences of actions as models of computations up to concurrency are replaced by *partial orders* of actions such that the set of the linearizations of each partial order exactly corresponds to an equivalence class of totally ordered sequences. In this way, each Mazurkiewicz trace gives rise to a partially ordered multiset (pomset) of actions, i.e. a labeled partial order up to isomorphism. Indeed, the elements of each Mazurkiewicz trace are in bijective correspondence to the linearizations of such a labeled partial order.

In the theory of Petri nets, firing sequences of transitions (in a given net) play the role of action sequences. The causal dependencies between the occurrences of the transitions can be faithfully captured by a (deterministic) occurrence net [Goltz and Reisig, 1983] such that all possible firing sequences of the occurrence net are equivalent to the original firing sequence. It is also possible to compose such occurrence nets, and in this context one speaks of concatenable processes [Sassone, 1996]. Finally, the idea of concatenable processes as a model of computation paths up to concurrency has been successfully applied to graph transformation systems [Baldan et al., 1998a]. The latter work introduces occurrence grammars, which are the graph rewriting analogon of occurrence nets.

The purpose of Section 4 is to provide such a theory of concatenable processes also for double pushout rewriting in adhesive categories. Preliminary

[7] The proofs of [Bonchi and Heindel, 2006] only apply to weakly adhesive categories with effective unions [Barr, 1987].

results have been published in the author's [Baldan et al., 2006a] and the notion of causality has been further analyzed in [Corradini et al., 2008]. Again, one can establish a bijective correspondence between processes as partial order representations of computations and switch-equivalence classes of rewriting sequences. In fact, it is even possible to obtain this correspondence in any weakly adhesive category and not only in adhesive categories as described in [Baldan et al., 2006a].

Section 3 and Section 4 discuss two different models of finite computation paths up to concurrency, which generalize the ideas of Mazurkiewicz traces and pomsets, respectively. Mazurkiewicz traces and pomsets are equivalent models of the same phenomenon; also their generalizations are equivalent (see Theorem 4.14).

Section 5 and Section 6 will finally apply the idea of a concurrency respecting model of computations to the set of *all* (possibly conflicting) computation paths that originate from a given start state. The procedure which generates such a model of all possible evolutions of a given system has become known as unfolding [Esparza and Heljanko, 2008]; the result of unfolding is a (possibly infinite) structure which is referred to as the unfolding of the system (from a given state).

Section 5 begins the discussion of unfoldings for systems that have finite transition systems. In this special case it becomes evident in what sense unfoldings are much more compact than explicit transition systems: they are exponentially smaller. At the same time, unfoldings contain explicit information concerning the causal dependencies of all possible events of a system. This fact has been exploited by McMillan for the verification of systems [McMillan, 1995]. The latter work also describes a very convenient algorithm for the construction of unfoldings of Petri nets. One could try to describe the general idea of the unfolding algorithm as follows.

Start by considering all actions that could occur immediately in the start state. For each of these "immediate" action occurrences, temporarily mark the resources in the start state that would be deleted or *consumed* by the action; moreover adjoin the resources that would be created or *produced*. Label the marked part and the new resources with the action occurrence. Already after this first unfolding step, several possible action occurrences can be in conflict.

After this initial step, further unfolding steps are executed repeatedly – "ad infinitum". In each of these general unfolding steps and for each new action occurrence that might be added, it is necessary to check whether the consumed resources are *concurrent*, which ensures that the resources actually

can be produced by a sequence of previously added action occurrences. This preliminary step is necessary because of possible conflicts and bifurcations. If the consumed resources are concurrent, the new action occurrence is actually added as before. Finally, the totality of all action occurrences that are obtained in this way together with the union of all (instances of) resources constitutes the *full* unfolding of the system. As a characteristic feature, each action occurrence has (usually unique) minimal causal explanations and causality is acyclic. Moreover, for reasons of efficiency, it is desirable to have a *static* description of concurrency of resources.

The first task in Section 5 is the description of this unfolding algorithm for systems in which resources are modeled as objects of weakly adhesive categories and action types correspond to rewrite rules. This is done in analogy to the description of the unfolding algorithm in [McMillan, 1995]. With suitable assumptions on the categories in which rewriting takes, the full unfolding can be constructed. These full unfoldings are actually *complete* by construction: every run of the original system is represented in the unfolding – together with causality information.

For a practical, efficient unfolding procedure it is desirable to have a static method to determine such *concurrent* resources. Hence, from a practical point of view, besides completeness, an important result of Section 5 is that unfolding can be performed statically, i.e. concurrency of resources can be checked without the need to solve reachability problems.

Applications of the general unfolding technique to system analysis and verification abound (see [Esparza and Heljanko, 2008] for an overview). In the area of the analysis and verification of dynamic, distributed systems, in which the structure of states is represented by graphs or graph-like objects, the work [Baldan et al., 2004] on finite systems shows how existing techniques for Petri nets can be applied to graph transformation systems. Moreover, these techniques can also be applied to infinite state systems by safe over-approximations of systems which contain additional states and transitions [Baldan et al., 2008c]. In fact, there are promising case studies based on implementations (see [König and Kozioura, 2006a]).

The theory on which the above mentioned applications are based works with (hyper-)graphs as a means to model the structure of system states. The main purpose of Section 5 is to extend the theory and to make the unfolding technique available for systems which call for more complicated models, such as graphs with attributes [Ehrig et al., 2004c], scopes and general information concerning the topological nature of states. In this respect, the present thesis is similar

to the monograph [Ehrig et al., 2006]. The latter gives generalized, abstract versions of the "classical" results of double pushout graph transformation that are reviewed in the handbook [Corradini et al., 1997], including a Church-Rosser theorem for parallel-independent rewriting steps and commutativity of sequential-independent ones (see also [Habel et al., 2001]). The present thesis complements this monograph with a theory of processes and unfoldings.

The main theoretical challenge of the first part of the thesis is addressed in Section 6, which concerns the category theoretical characterization of unfoldings as proposed by Winskel [Nielsen et al., 1981]. In the same way as there is the free (term) algebra for every signature, the unfolding of a set of rewriting rules and a start object is a (co-)free structure. This result has been published in the author's [Baldan et al., 2009]. More precisely, the unfolding is the co-free *occurrence grammar* of a given *grammar* – a set of rules with a start object. Thus, the unfolding is a functor which is right adjoint to the embedding of the category of occurrence grammars into the category of grammars.

That unfolding is a co-free construction is the main result of the first part on the concurrent semantics of rewriting. It generalizes and improves previous results on graph rewriting [Baldan et al., 1998b, Baldan et al., 2007]. A crucial contribution is a novel notion of grammar morphism which is devised in analogy to the Petri net morphisms of [Nielsen et al., 1981]. Further – using a certain semantics for Petri nets – the main result even can be seen as a solution to the longstanding problem that unfolding of general Petri nets cannot be be characterized exactly as a co-free construction (see [Hayman and Winskel, 2008]). Finally, this result is not only of theoretical interest: it establishes unfoldings as a compositional analysis and verification technique, as the unfolding of a composed system can be obtained by composing the unfoldings of the components (see [Baldan et al., 2006b, Baldan et al., 2008b]).

A characteristic feature of the results of the first part of this thesis is that they are *not* developed on the concrete level of a number of example categories of graphs that are used in various areas of computer science. Instead, they are established for any suitable (abstract) category which shares a small list of fundamental properties with the category of graphs; e.g. it is possible to work in any (quasi-)topos [Johnstone, 1977, Wyler, 1991] or any adhesive category [Lack and Sobociński, 2005].

The second part of this thesis is a study of the "essential" properties of the category of graphs and provides the category theoretical foundations of the results of the first part. An overview of its constituent sections is given in Figure 13. First, Section 7 discusses the general category theoretical framework

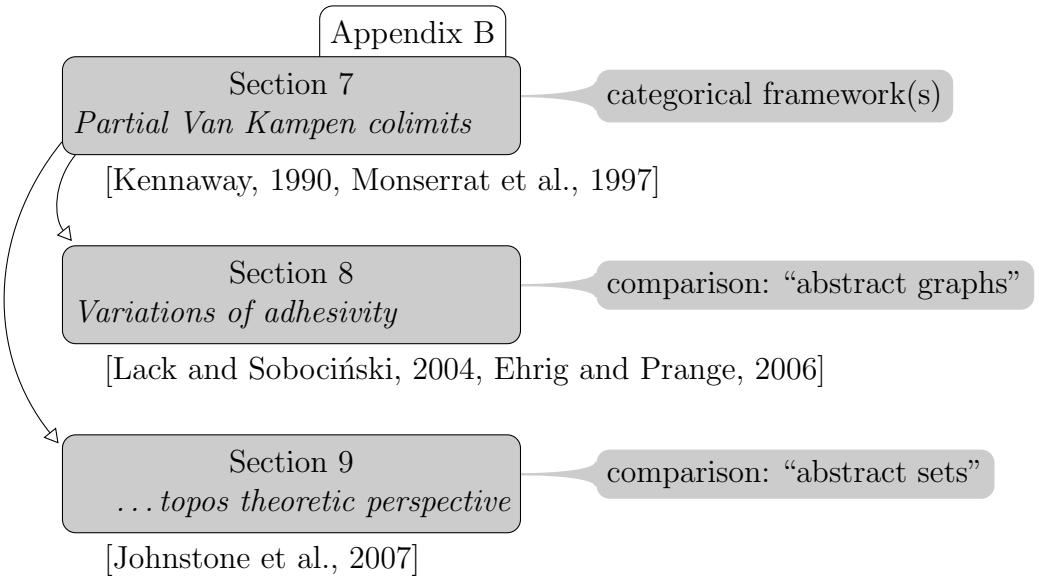


Figure 13: Structure of the second part *Categorical Foundation* and some of its related work

in which the results of the first part are derived. The two following sections compare this framework with existing approaches to capture fundamental properties of the categories of graphs and sets, respectively. Section 8 discusses the relation to two classes of "graph-like" categories, namely adhesive categories [Lack and Sobociński, 2004] and a recent variation of them which has been used in the area of graph transformation systems [Ehrig and Prange, 2006]. Finally, Section 9 makes the connection to the well-established notion of a topos, which one might want to think of as an "abstract set" or a generalized topological space.

Section 7 introduces the theory behind the metaphorical compatibility of clipping and gluing that has been described in Section 1.3.2 – not only for pushouts but for all colimits. The motivation is that the category of graphs does not only have all sums and quotients (a fact which allows to construct all colimits) but in fact these colimits are "well-behaved" in the following sense: they are stable w.r.t. the embedding into the category of graphs and *partial* graph morphisms. This property alone allows to derive the major part of the fundamental properties of pushouts and colimits that are used in the first part of the thesis. Hence the main focus of Section 7 is on colimits with this

property; they are dubbed *partial* Van Kampen colimits.

The idea to consider categories of partial morphisms is already present in early work on the foundations of single pushout rewriting [Kennaway, 1990] (see also [Monserrat et al., 1997]). In a different direction, "proper" Van Kampen colimits [Heindel and Sobociński, 2009] can often be obtained from partial ones by the additional assumption of pullback stability (see Appendix B).

Section 8 then shows how "stable gluing" as formalized by partial Van Kampen pushouts is related to previous definitions of "graph-like" categories, namely adhesive categories [Lack and Sobociński, 2004] and weak adhesive high-level replacement categories [Ehrig and Prange, 2006]. As a fact, pushouts in adhesive categories satisfy a properly stronger property (since they moreover are stable under pullback). The case for weak adhesive high-level replacement (HLR) categories is slightly more complicated; however, making suitable assumptions about *images* of morphisms, every *weak* Van Kampen square in the sense of [Ehrig and Prange, 2006] is a partial Van Kampen pushout. The main goal of Section 8 is to make precise in what sense the condition on pushouts in adhesive categories is stronger than necessary and to provide a more conceptual perspective on the rather complicated definition of weak adhesive high-level replacement categories.

Finally, Section 9 takes the perspective of topos theory [Johnstone, 2002]. That topos theory is relevant for the theory of adhesive categories has already been stated by the inventors of adhesive categories [Lack and Sobociński, 2006, Johnstone et al., 2007]. For the present thesis, the main fact is that in every topos, all colimits are partial Van Kampen and moreover stable under pullback. This can be taken to mean that everything that has been known to be "set-like" for a long time is also "graph-like". Moreover, topos theory provides a large supply of "graph-like" structures that can be used to model system states.

[Baldan et al., 2006a] Baldan, P., Corradini, A., Heindel, T., König, B., and Sobocinski, P. (2006a). Processes for adhesive rewriting systems. In [Aceto and Ingólfsdóttir, 2006], pages 202–216

[Baldan et al., 2009] Baldan, P., Corradini, A., Heindel, T., König, B., and Sobociński, P. (2009). Unfolding grammars in adhesive categories. In *Proceedings of CALCO 2009 (Algebra and Coalgebra in Computer Science)*. Springer. To appear

[Heindel, 2008] Heindel, T. (2008). Grammar morphisms and weakly adhesive categories. In [Ehrig et al., 2008], pages 493–495

[Heindel and Sobociński, 2009] Heindel, T. and Sobociński, P. (2009). Van Kampen colimits as bicolimits in Span. In *Proceedings of CALCO 2009 (Algebra and Coalgebra in Computer Science)*. Springer. To appear

[Bonchi and Heindel, 2006] Bonchi, F. and Heindel, T. (2006). Adhesive DPO parallelism for monic matches. In *Graph Transformation for Verification and Concurrency, GT-VC* 2006

[Bonchi et al., 2008] Bonchi, F., Gadducci, F., and Heindel, T. (2008). Parallel and sequential independence for borrowed contexts. In [Ehrig et al., 2008], pages 226–241

[Corradini et al., 2006] Corradini, A., Heindel, T., Hermann, F., and König, B. (2006). Sesqui-pushout rewriting. *Lecture Notes in Computer Science*, 4178:30–

[Baldan et al., 2005] Baldan, P., Corradini, A., Esparza, J., Heindel, T., König, B., and Kozioura, V. (2005). Verifying red-black trees. In *Proceedings of COSMICAH '05*. Proceedings available as report RR-05-04 (Queen Mary, University of London)

[Heindel, 2009] Heindel, T. (2009). Towards secrecy for rewriting in weakly adhesive categories. *Electronic Notes in Theoretical Computer Science*, 229(3):97–115

[Baldan et al., 2008a] Baldan, P., Bonchi, F., Heindel, T., and König, B. (2008a). Irreducible objects and lattice homomorphisms in adhesive categories. In Pfalzgraf, J., editor, *Proceedings of ACCAT '08 (Workshop on Applied and Computational Category Theory)*

Table 1: List of publications by the author

2 *Background on single and double pushout rewriting*

This section presents a review of (the preliminaries about) the two main approaches to rewriting that the investigation of this thesis is based on, namely the *double pushout approach* (DPO), which was introduced in [Ehrig et al., 1973], and the *single pushout approach* (SPO), originally proposed in [Raoult, 1984] and developed in [Löwe and Ehrig, 1990]. An overview of, and a comparison between DPO and SPO rewriting of *graphs* can be found in [Rozenberg, 1997, Corradini et al., 1997, Ehrig et al., 1997]. In general, both single and double pushout rewriting can be used to define a transformation relation on the objects of a given category[8].

As a historic fact, the main focus of single and double pushout rewriting was on concrete categories such as the category of *(multi) graphs* (see Definition A.1) or *term graphs* (see e.g. [Plump, 1997]). Over time, the focus of research kept shifting from concrete categories towards *abstract* categories: whatever the objects of rewriting are, they merely have to come equipped with an appropriate notion of *morphism* between them. In particular, objects are not required to be sets with structure and morphisms are not necessarily structure preserving functions. The crucial point is that independed of the concrete nature of objects and morphisms, the resulting *category* can serve as a rewriting framework, provided that it satisfies certain properties.

The early frameworks for SPO and DPO rewriting in abstract categories in [Ehrig et al., 1990, Kennaway, 1990, Ehrig and Löwe, 1993, Padberg, 1993] are based on a number of properties which are known to hold for graph morphisms and feature prominently in the proofs of the "standard" theorems of graph transformation (cf. the overview in [Habel et al., 2001]). However the diversity of the conditions that one has to stipulate to obtain the desired theorems is rather discouraging (see [Padberg, 1993]).

The *adhesive categories* of [Lack and Sobociński, 2005] provide a solution to this problem. Inspired by the work of [Brown and Janelidze, 1997] on generalized versions of the Seiffert-Van Kampen theorem of topology, the work [Lack and Sobociński, 2005] describes a *single* property of pushouts that allows to develop representative parts of the theory of double pushout graph transformation in an abstract setting. This single property is studied systematically in [Heindel and Sobociński, 2009].

The recent work [Ehrig et al., 2004b, Ehrig and Prange, 2006] generalizes

[8]The preliminaries about categories and notational conventions are summarized in Appendix A.

the adhesive categories of Lack and Sobociński to capture a properly larger class of structures (which are used in the area of software engineering). It introduces (weak) adhesive high-level replacement categories, which encompass additional, common structures of computer science which do not fit the framework of (quasi-)adhesive categories, such as simple graphs (cf. [Johnstone et al., 2007]) and other variants of (hyper-)graphs (cf. [Ehrig et al., 2006]).

The present thesis follows this line of research and is based on a variation of adhesive categories that is proposed in [Heindel, 2008]. The latter work tries to combine the theoretical elegance of adhesive categories and the generality of the weak adhesive high-level replacement categories.

Overview of the section. First, the two main approaches to rewriting are reviewed, namely the single pushout approach (Section 2.1) and the double pushout approach (Section 2.2). Besides the (very succinct) formal definitions of rewriting steps – the so-called *direct derivations* – according to these two approaches, a variety of concrete examples is given.

With the definitions of the basic rewriting formalisms at hand, Section 2.3 reviews how concurrency is modelled in double pushout rewriting. The two central definitions are the *sequential* and *parallel* independence of pairs of consecutive and branching rewriting steps, respectively. Moreover, two standard theorems of double pushout graph rewriting are presented, namely the Church-Rosser Theorem and the sequential commutativity theorem, which formally capture the interplay between parallel and sequential independence. As before, the definitions and theorems are illustrated with concrete examples. Thus, Section 2.3 covers the essential prerequisites for the results of the first part on the concurrent semantics of rewriting.

Next, Section 2.4 sets out with the definition of adhesive categories and further discusses (some of) their properties which allow to derive standard theorems of graph rewriting in an abstract, category theoretical setting. As a paradigmatic example, the sequential commutativity theorem is treated in detail; particular attention is given to the fact that its proof describes a procedure to reschedule a pair of consecutive, sequential-independent rewriting steps. Whenever the need arises, Section 2.4 reviews the relevant concepts that go beyond the standard repertoire of basic category theory[9].

Most of the material in this section applies to double pushout rewriting. However, as made precise in Section 2.5, double pushout rewriting is a special

[9]As mentioned before, the preliminaries about categories and notational conventions are summarized in Appendix A.

case of single pushout rewriting if single pushout rewriting is performed in categories of partial maps (and a certain rule format is used). Then each double pushout rewriting step in a category is nothing else but a reversible single pushout step in the partial map category. Hence, the central concept of Section 2.5 are partial maps in abstract categories. A digression concerning the role of partial map classifiers is added as the latter will feature prominently in the second part of this thesis on the categorical foudations. The proof of the main fact, namely that *single pushout* "subsumes" *double pushout*, makes use of a (variant of the) pullback complements of [Dyckhoff and Tholen, 1987], which are thus discussed as well. In summary, Section 2.5 describes the conceptual similarity of single and double pushout rewriting.

Finally, to put adhesive categories into perspective, Section 2.6 describes two generalizations that have been proposed in the literature, namely quasi-adhesive categories [Lack and Sobociński, 2005] and weak adhesive high-level replacement categories [Ehrig and Prange, 2006]. The latter approach is more general, however at the price of a more involved definition. In turn, as described in [Johnstone et al., 2007], the main drawback of the framework of quasi-adhesive categories is that it fails to capture the archetypical simple graphs (see Example 2.37), which occurr passim in computer science. The latter work established a tight relation with the (regular) effective unions of [Barr, 1987]. Hence, also the latter are discussed even though they have turned out to be only of minor importance for the present thesis. The role of Section 2.6 is to provide the vocabulary for a discussion about convenient formulations of the properties of graphs that allow to develop the existing theory of graph rewriting category theoretically.

 Remark 2.1 Taken in isolation, none of the facts that will be mentioned in this section are original contributions in the strict sense, though some of them are not easily found in the existing literature – at least not in the form presented here – or are folklore (outside of mainstream computer science). However, the presented facts have never been assembled together under the heading of single *and* double pushout rewriting in *abstract* categories. Especially the emphasis on the role of partial map categories for the investigation of the properties of (variations of) adhesive categories is only shared by [Cockett and Guo, 2007], which is not concerned with rewriting. Moreover, the use of the category of topological spaces as one of the *paradigmatic* examples is a novelty.

For the remainder of this section, let $\mathbb{C}$ be a fixed category, and all mentioned objects and arrows are assumed to belong to $\mathbb{C}$, unless stated otherwise. All (basic) notions and results from category theory that will be used are listed

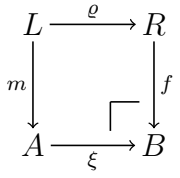
in Appendix A; however the first 30 pages of [Pierce, 1991] might be a better starting point for category theory novices. Some acquaintance with topology might help understand certain examples but is not necessary.

2.1 Single pushout approach

Though conceived later in the history of the algebraic approaches to rewriting, the single pushout (SPO) approach is conceptually simpler and thus will be used to introduce the general idea of rewriting in category theoretical terms. It seems that at least Van den Broek would agree with this choice, as he writes that it "would conceptually be much simpler if single pushouts could be used instead of double pushouts" [van den Broek, 1991]. The simple reason is that the definition of the SPO approach is more succinct than the one of the double pushout approach (cf. Section 2.2) since it uses only one pushout square instead of two.

Given a morphism in a category, the SPO approach associates it with a rewriting relation over objects as follows: an object can be transformed to another one using the morphism if it can be obtained by taking the pushout along the given morphism. The details are given in the following definition, which is based on [Ehrig and Löwe, 1993].

☼ Definition 2.2 (Single pushout rewriting) Let $\mathbb{M}$ be a lluf subcategory of $\mathbb{C}$, i.e. $\mathbb{M}$ is a subcategory $\mathbb{M} \subseteq \mathbb{C}$ with the same collection of objects $\mathrm{ob}(\mathbb{M}) = \mathrm{ob}(\mathbb{C})$. Let $\varrho\colon L \to R$ in $\mathbb{C}$ be a morphism, which in this context is referred to as the *rule*, and let $m\colon L \to A$ in $\mathbb{M}$ be another one, which is called a *match* or *redex* for the rule ϱ.



Then ϱ *rewrites* A *at* m *to* B, written $A \models\!\!\!\Rightarrow_{\langle\varrho,m\rangle} B$ or also $B \Leftarrow\!\!\!\models_{\langle\varrho,m\rangle} A$, if B is a pushout object of $A \leftarrow_m - L -_\varrho \rightarrow R$ in the category $\mathbb{C}$, i.e. if there exists a pushout $A -_\xi \rightarrow B \leftarrow_f - R$ of $A \leftarrow_m - L -_\varrho \rightarrow R$ as displayed. The resulting pushout square is called an SPO *direct derivation diagram* in this context, and the morphism $f\colon R \to B$ is called the *back-match* of the derivation diagram. The rule ϱ *rewrites* A *to* B, written $A \models\!\!\!\Rightarrow_\varrho B$ or $B \Leftarrow\!\!\!\models_\varrho A$, if there exists some morphism $m\colon L \to A$ in $\mathbb{M}$ such that $A \models\!\!\!\Rightarrow_{\langle\varrho,m\rangle} B$ holds. In this situation, the tuple $\langle A, m, \varrho, B\rangle$ or – to enhance readability – also the expression '$A \models\!\!\!\Rightarrow_{\langle\varrho,m\rangle} B$' is called an SPO *rewriting step*. The pushout square which is displayed above is said to *witness* the derivation step $A \models\!\!\!\Rightarrow_{\langle\varrho,m\rangle} B$. ☼

A direct consequence of this definition is that the result of rewriting an object A with a rule $\varrho\colon L \to R$ at a match $m\colon L \to A$ is determined up to

a canonical isomorphism. More precisely, let B and C be objects such that $B \Leftarrow\!\langle\varrho,m\rangle\!\!\dashv A$ and $A \vDash\!\langle\varrho,m\rangle\!\!\Rightarrow C$ hold; then B and C are isomorphic, and the mediating morphism between the two pushout objects B and C is the canonical isomorphism.

Example 2.3 (Set rewriting) Take for both $\mathbb{C}$ and $\mathbb{M}$ the category $\mathbb{Sets}$, which has sets as objects and functions as morphisms, i.e. $\mathbb{C} = \mathbb{M} = \mathbb{Sets}$.

$$\begin{array}{ccc} \{a,b\} & \xrightarrow{\varrho} & \{c\} \\ {\scriptstyle m}\downarrow & & \downarrow \\ \{d\} & \longrightarrow & \{d\} \end{array}$$

Now consider the rule $\varrho\colon \{a,b\} \to \{c\}$ with the function $m\colon \{a,b\} \to \{d\}$ as match. Then the rule ϱ rewrites $\{d\}$ at the match m to $\{d\}$ (or any other singleton set $\{\star\}$); hence rewriting via ϱ does not necessarily change the rewritten object. The result of rewriting crucially depends on the involved match. To ensure that the rule ϱ always changes rewritten objects, one could require matches to be monic, which amounts to taking the category of sets and injective functions for $\mathbb{M}$.

A (variant of) the token game for Petri nets is obtained if rewriting is performed in categories of *colored sets* and *color preserving (partial) functions*. The fundamental ideas is as follows: in a Petri net with a set of places P, each marking is considered as a P-colored set, i.e. a set with an assignment which maps each of its elements to one of the colors in P. Further each transition can be thought of as a color preserving, partial function (with empty domain of definition). This is made more explicit in the following example.

Example 2.4 (Petri nets via colored set rewriting) A Petri net in the sense of [Winskel, 1985] is a triple $\mathcal{N} = \langle P, T, \mathrm{pre}, \mathrm{post}\rangle$ where P and T are disjoints sets of *places* and *transitions*, respectively, and $\mathrm{pre}, \mathrm{post}\colon T \to P^{\oplus}$ give for each transition $t \in T$ its *pre-* and *post-set*, respectively, where $P^{\oplus}$ is the free commutative monoid of *multisets*[10] *over* P and its elements $\mu \in P^{\oplus}$ are functions $\mu\colon P \to \mathbb{N}$ with domain P and the natural numbers as codomain.

Each marking $\mu \in P^{\oplus}$ can be represented by a total function with codomain P, say $[\![\mu]\!]\colon P \otimes \mu \to P$ where $P \otimes \mu$ contains exactly $\mu(p)$ copies of each $p \in P$, e.g. $P \otimes \mu = \bigcup\{\{p\} \times \{\!|\mu(p)|\!\} \mid p \in P\} \subseteq P \times \mathbb{N}$ where $\{\!|\mu(p)|\!\} = \{0, \ldots, \mu(p) - 1\}$, and the function $[\![\mu]\!]$ maps each $\langle p, i\rangle \in P \otimes \mu$ to p, i.e. $[\![\mu]\!]$ is the projection to P; an example is given in Figure 14. Each transition $t \in T$ gives rise to a nowhere defined partial map $[\![t]\!]\colon (P \otimes \mathrm{pre}\, t) \rightharpoonup (P \otimes \mathrm{post}\, t)$,

[10] Let $P \in \mathbb{Set}$ be a finite set. The free commutative monoid $P^{\oplus} = \langle \mathbb{Set}(P, \mathbb{N}), \oplus, \underline{0}\,\rangle$ has all functions $\mu\colon P \to \mathbb{N}$ as elements, the binary operation $\oplus$ is pointwise addition, and the neutral element $\underline{0}$ is the constant 0-function. This means that $(\mu \oplus \nu)p = \mu(p) + \nu(p)$ and $\underline{0}(p) = 0$ for all $p \in P$.

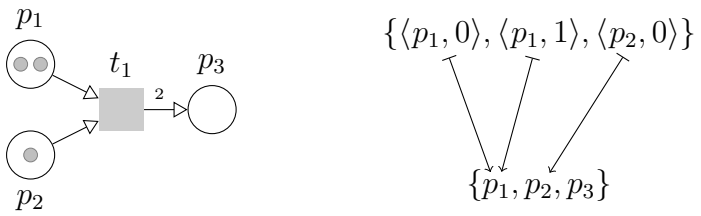


Figure 14: A marked Petri net and its marking as a colored set

which hence "preserves" the coloring of elements. An example of the encoding of a rule is illustrated in Figure 15. On the left in the figure, a transition is

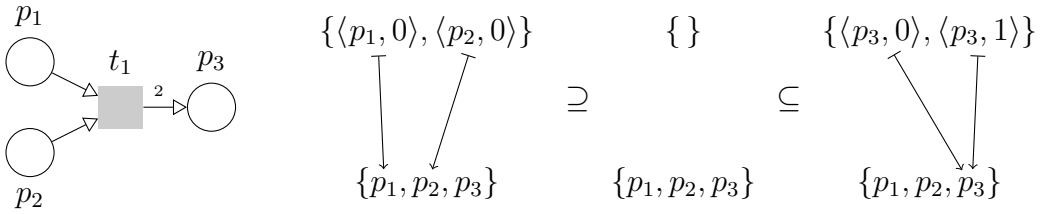


Figure 15: Encoding a Petri net transition as a rule of colored sets

shown which has two places p_1, p_2 in its pre-set and twice the place p_3 in its post-set; on the right, the corresponding partial map is given by a pair of inclusions from the empty set.

In this way, each Petri net $\mathcal{N} = \langle P, T, \mathrm{pre}, \mathrm{post}\rangle$ can be encoded by a set of rules $[\![\mathcal{N}]\!]$, namely $[\![\mathcal{N}]\!] := \{[\![t]\!] \mid t \in T\}$.[11] Moreover, for each step $(\mu \oplus \mathrm{pre}\, t)\ [t\rangle\ (\mu \oplus \mathrm{post}\, t)$ in the net there is a corresponding single pushout derivation $[\![\mu \oplus \mathrm{pre}\, t]\!] \models\!\!\!\Rightarrow_{[\![t]\!]} [\![\mu \oplus \mathrm{post}\, t]\!]$.

In the definition of single pushout rewriting, the condition that each match $m\colon L \to A$ belongs to some subcategory $\mathbb{M} \subseteq \mathbb{C}$ can be used to ensure certain properties of the rewriting relation, or that pushouts along matches always exist. As a typical example, take the category of partial graph morphisms for $\mathbb{C}$ and instantiate $\mathbb{M}$ with the category of graphs and (total) graph morphisms. Then, according to [Löwe and Ehrig, 1990], using only total morphisms as

[11] If the net T contains a doubleton $\{t_1, t_2\}$ such that $\langle \mathrm{pre}\, t_1, \mathrm{post}\, t_1\rangle = \langle \mathrm{pre}\, t_2, \mathrm{post}\, t_2\rangle$, the encoding of a transition t should be changed to $\{t\} \times [\![t]\!]\colon \{t\} \times (P \otimes \mathrm{pre}\, t) \rightharpoonup \{t\} \times (P \otimes \mathrm{post}\, t)$. This encoding of nets as rule sets still assumes the standard condition that there are no transitions with empty pre- *and* post-sets, i.e. either $\mathrm{pre}\, t \neq \underline{0}$ or $\mathrm{post}\, t \neq \underline{0}$ is required.

matches ensures that the resulting rewriting relation is sufficiently intuitive[12]. As an example of a possibly counter-intuitive rewriting relation, consider the following rewriting step in the category of partial functions.

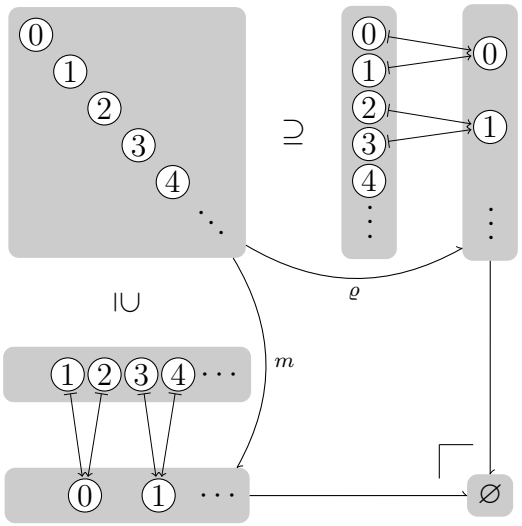


Figure 16: A single pushout rewriting step in the category of partial functions

Example 2.5 (Rewriting with partial functions as matches) Let $\mathbb{P}\mathfrak{fn}$ be the category with sets as objects and partial functions as morphisms, and take $\mathbb{C} = \mathbb{M} = \mathbb{P}\mathfrak{fn}$. Now consider the direct derivation diagram illustrated in Figure 16. The rule $\varrho\colon \mathbb{N} \rightharpoonup \mathbb{N}$ and the match $m\colon \mathbb{N} \rightharpoonup \mathbb{N}$ are given by

$$\varrho\colon \mathbb{N} \rightharpoonup \mathbb{N} \qquad n \mapsto \left\lfloor \tfrac{n}{2} \right\rfloor \qquad\qquad m\colon \mathbb{N} \rightharpoonup \mathbb{N} \qquad n \mapsto \begin{cases} \text{undefined} & \text{if } n = 0 \\ \left\lfloor \frac{n-1}{2} \right\rfloor & \text{otherwise} \end{cases}$$

where $\mathbb{N} \in \mathbb{P}\mathfrak{fn}$ is the set of natural numbers. The result of rewriting the set $\mathbb{N}$ at match m using the rule ϱ is the empty set $\varnothing$.

The third example for SPO-rewriting will describe a way to mimic word rewriting à la Chomsky; it is mainly meant to hint at the versatility of single

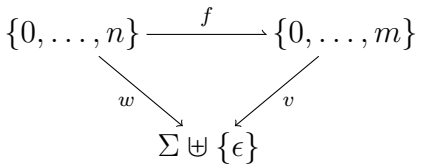
[12] "In this framework, we could define redices as partial morphisms as well. Although technically easier this choice would lead to a counter-intuitive expressive power of the rules. Since the empty morphism [...] is a partial graph morphism, it is a redex of every rule in arbitrary graphs. This results in every rule being applicable in every situation [...]. Therefore, we stick to the conventional approach for derivations in rule-based systems, which requires a total match of the rule's left hand side as application condition." [Löwe and Ehrig, 1990]

pushout rewriting. It is not exactly rewriting on words, but uses a slightly generalized notion of word to obtain a succinct presentation using the SPO-approach.

Example 2.6 (Encoding word rewriting) To define word rewriting à la Chomsky by means of SPO (or DPO) rewriting, it is technically convenient to consider ϵ-*stuffed* words: a new alphabet symbol ϵ is introduced and any "real" symbol in a word is embraced by two adjacent ϵ's. For example *abab* over $\Sigma = \{a, b\}$ would become $\epsilon a \epsilon b \epsilon a \epsilon b \epsilon$, and the empty word $\lambda \in \Sigma^*$ corresponds to the (non-empty) ϵ-stuffed word $\epsilon \in \Sigma \uplus \{\epsilon\}$.

In this way, words over Σ become very similar to (a pointer to) a linked list of characters as known from computer programming: the first ϵ corresponds to the pointer to the list, the last ϵ is the `nil`-pointer, and each letter/ϵ pair can be thought of as two adjacent memory cells, say $\boxed{a}\boxed{\bullet}\!\!\rightarrow$, which contain a character and a pointer to the next (proper) character; finally two consecutive pointers $\boxed{\bullet}\!\!\rightarrow\boxed{\bullet}\!\!\rightarrow$ are the counterpart of two consecutive ϵ's.

To some extent, this description of words over Σ as ϵ-stuffed words is analogous to the representation of arrays of characters as linked lists of characters. The advantage of linked list consists in avoiding the need to copy parts of the array whenever elements are deleted from or inserted into the array, and the reason for why ϵ-stuffed words appear to be more suitable for the description of word rewriting using the single pushout might be found by exploring this analogy.



To give the concept of ϵ-stuffed word a formal counterpart, one could define them as $\Sigma \uplus \{\epsilon\}$-labelled, non-empty, finite ordinals, which means that each word is a function $w\colon \{0, \ldots, n\} \to \Sigma \uplus \{\epsilon\}$.

To obtain a category, take the order and label preserving partial functions as morphisms: a morphism between words $w\colon \{0, \ldots, n\} \to \Sigma \uplus \{\epsilon\}$ and $v\colon \{0, \ldots, m\} \to \Sigma \uplus \{\epsilon\}$ is a partial function $f\colon \{0, \ldots, n\} \rightharpoonup \{0, \ldots, m\}$ which is monotone and satisfies the equation $w = v \circ f$. The resulting category is referred to as $\mathbb{W}\mathrm{ords}_\Sigma$.

Now let $G = \langle N, \Sigma, P, S\rangle$ be a grammar and let $\langle \alpha, \beta\rangle \in P$ be a production. Then there are the obvious ϵ-stuffed variants $\alpha'\colon \{0, \ldots, n\} \to N \cup \Sigma \uplus \{\epsilon\}$ and $\beta'\colon \{0, \ldots, m\} \to N \cup \Sigma \uplus \{\epsilon\}$ where $n = 2|\alpha|$ and $m = 2|\beta|$. The encoding of the production of $\langle \alpha, \beta\rangle$ is the rule $\varrho\colon \{0, \ldots, n\} \rightharpoonup \{0, \ldots, m\}$ which is everywhere undefined except for $\varrho(0) = 0$ and $\varrho(n) = m$, and this is actually a morphism in $\mathbb{W}\mathrm{ords}_{N \cup \Sigma}$.

Next take $\mathbb{C} = \mathbb{W}\mathtt{ords}_{N\cup\Sigma}$ as the category in which rewriting takes place and use the subcategory of *word embeddings* as morphisms for $\mathbb{M}$ where a word embedding is a total function $f\colon \{0,\dots,n\} \to \{0,\dots,m\}$ satisfying $f(i+1) = f(i)+1$ for all natural numbers $i < n$. Now a string γ can be derived to δ using the production $\langle \alpha, \beta\rangle$ alone, i.e. $\gamma = \varphi\alpha\psi$ and $\delta = \varphi\beta\psi$ hold for some pair φ and ψ, if and only if $\gamma' \models_{\varrho}\Rightarrow \delta'$ holds for the obvious ϵ-stuffed variants of γ and δ, respectively. As a final remark, if $\gamma' \models_{\langle\varrho,f\rangle}\Rightarrow \delta'$ holds for some concrete match f, then f carries all information about the involved words ψ and φ.

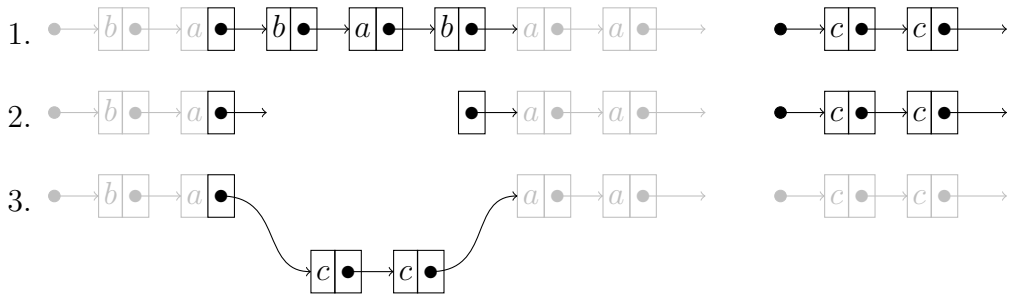


Figure 17: Applying the production $bab \to cc$ in three steps using linked lists

Summarizing, as illustrated in Figure 17, rewriting in $\mathbb{W}\mathtt{ords}_{N\cup\Sigma}$ mimics the usual algorithm for replacing a sub-list by a (copy of) another list. After identifying (and removing) an occurrence of the pattern $\epsilon b\epsilon a\epsilon b\epsilon$ in the larger list $\cdots\epsilon b\epsilon a\epsilon b\epsilon a\epsilon b\epsilon a\epsilon a\epsilon\cdots$, the pointer into the occurrence of the pattern becomes the pointer into the replacement list $\epsilon c\epsilon c\epsilon$, and the pointer out of the occurrence of the pattern is used as the pointer out of the replacement list. ⌾

Despite the aforementioned conceptual simplicity of single pushout rewriting, the most well-known approach to rewriting is nevertheless the double pushout approach. However, for the case of single pushout rewriting in "well-behaved" categories of partial morphism using monic matches, the two approaches are similar in nature, and, in the context of this thesis, their differences are rather a technical detail than a major concern.

2.2 Double pushout approach

The double pushout (DPO) approach has been proposed prior to the single pushout approach in [Ehrig et al., 1973]. One characteristic of DPO-rewriting

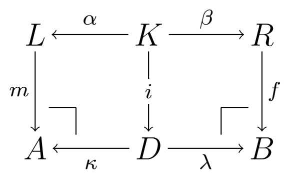
consists in the fact that each atomic rewriting step is divided into two phases, which in many applications naturally correspond to the consumption and generation of resources, respectively. However, as a direct consequence of Fact 2.33 below, the DPO-approach can be thought of as nothing else but a reversible variant of the single pushout approach, which ensures that for every rewriting step from A to B using a rule, there is a corresponding rewriting step from B to A in the opposite direction which applies the "opposite" rule at the back-match.

Rewriting following the DPO-approach in a category $\mathbb{C}$ uses $\mathbb{C}$-spans as rules. Derivation steps are defined by means of certain diagrams containing a pair of neighboring pushout squares – whence the name.

☼ DEFINITION 2.7 (Double pushout rewriting) A DPO *rule* q in $\mathbb{C}$ is a $\mathbb{C}$-span $q = L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ as shown in Figure 18(a). The three objects L, K, and R of q are *the components of* q and are called the *left-hand side*, the *gluing object*, and the *right-hand side*, respectively. The *opposite rule* of a rule $q = L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ is the mirrored span $q^\circ = R \xleftarrow{\beta} K \xrightarrow{\alpha} L$.

$$q = L \xleftarrow{\alpha} K \xrightarrow{\beta} R$$

(a) DPO rule



(b) DPO direct derivation diagram for q and m

Figure 18: DPO-rewriting

Let $q = L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ be a rule, and let $m\colon L \to A$ in $\mathbb{C}$ be a morphism; in this situation m is a DPO *match candidate* or *proto-match* for q. Further, a DPO *direct derivation diagram for* q *and* m is a diagram as shown in Figure 18(b), which comprises two pushout squares; the morphism $f\colon R \to B$ is called the *back-match* of the direct derivation diagram. A match candidate $n\colon L \to A$ for q is called a DPO *match for* q if there exists some direct derivation diagram for q and n.

In the situation of Figure 18(b), one says that *the rule* q *rewrites* A *to* B *at* m or B *is the result of rewriting the object* A *via the rule* q *at match* m, also written as $A \models\!\langle q,m\rangle\!\Rightarrow B$, and the direct derivation diagram in Figure 18(b) is said to *witness* that A can be rewritten to B via the rule q at match m. Moreover, the tuple $\langle A, m, q, B\rangle$ is referred to as a DPO *rewriting step* (though usually '$A \models\!\langle q,m\rangle\!\Rightarrow B$' is written instead of '$\langle A, m, q, B\rangle$'). ☼

As noted before, the rewriting step from A to B at m via $L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ as shown in Figure 18(b) is reversible since B can be rewritten to A at the back-match f using the opposite rule $R \xleftarrow{\beta} K \xrightarrow{\alpha} L$. As mentioned above, DPO rewriting in $\mathbb{C}$ could be defined as reversible SPO rewriting in this sense (if rule spans are assumed to consist of pairs of monic arrows and the category $\mathbb{C}$ suitably “graph-like”).Before delving into the details of the subtle issues of double pushout rewriting and the conditions for the existence of direct derivation diagrams, consider the following simple but paradigmatic example.

Example 2.8 (Set rewriting with injective functions) Take $\mathbb{C} = \mathbb{Sets}$ as the category in which rewriting takes place. Further let $q = L \overset{\alpha}{\leftarrowtail} K \overset{\beta}{\rightarrowtail} R$ be a span of injective functions and let $m\colon L \rightarrowtail A$ be another injection. Striving for simplicity, assume that these functions are actually inclusions, i.e. $L \supseteq K \subseteq R$ and $L \subseteq A$, and further suppose that $A \cap R \subseteq K$.

The rule q rewrites A at m to $(A \setminus (L \setminus K)) \cup (R \setminus K)$. This means that in a first step, the complement of K in L is removed from A (yielding $A \setminus (L \setminus K)$ as an intermediate result), and in a second step the complement of K in R, namely $R \setminus K$, is added, resulting in $(A \setminus (L \setminus K)) \cup (R \setminus K)$.

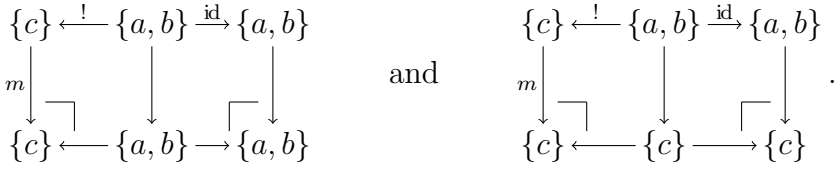
$$\begin{array}{ccccc} L & \longleftarrow & K & \longrightarrow & R \\ \downarrow & & \downarrow & & \downarrow \\ A & \longleftarrow & A \setminus (L \setminus K) & \longrightarrow & \begin{array}{c} R \setminus K \\ \cup \\ A \setminus (L \setminus K) \end{array} \end{array}$$

As a remark, the rule q corresponds to a partial function $\varrho_p\colon L \rightharpoonup R$ and the total function $m\colon L \rightarrowtail A$ can serve as a partial function as well. In fact, there is not only the DPO-rewriting step $A \overset{\langle q,m\rangle}{\Longrightarrow} (A \setminus (L \setminus K)) \cup (R \setminus K)$ in $\mathbb{Sets}$ but also an corresponding SPO-rewriting step $A \overset{\varrho_q}{\Longrightarrow} (A \setminus (L \setminus K)) \cup (R \setminus K)$ in the category of sets and partial functions (cf. Example 2.5).

In general, there are two central issues connected with the definition of double pushout rewriting, namely *existence* of direct derivation diagrams for a given rule $q = L \leftarrow K \rightarrow R$ and match candidate $m\colon L \rightarrow A$, and *uniqueness* of the result of rewriting. In fact, as shown in the next example, two rewriting steps $A \overset{\langle q,m\rangle}{\Longrightarrow} B$ and $A \overset{\langle q,m\rangle}{\Longrightarrow} C$ from the same object using the same rule-match pair do *not* necessarily result in isomorphic objects, i.e. $B \cong C$ does *not* hold in general.

Example 2.9 (Uniqueness of double pushout rewriting) Working again in $\mathbb{C} = \mathbb{Sets}$, take the rule $q = \{c\} \xleftarrow{!} \{a, b\} \xrightarrow{\mathrm{id}} \{a, b\}$ and the match $m\colon \{c\} \rightarrow \{c\}$. Then there are essentially two ways to construct a direct

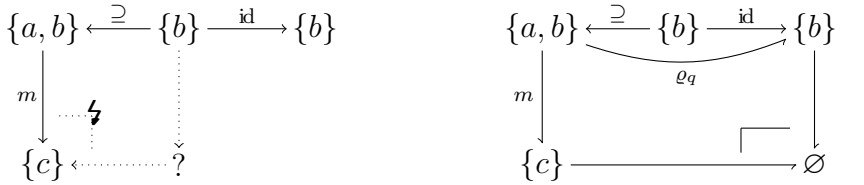
derivation diagram, namely



Hence both $\{c\} \models\!\!\!\sim_{\langle q,m\rangle} \{a,b\}$ and $\{c\} \models\!\!\!\sim_{\langle q,m\rangle} \{c\}$ hold, and clearly $\{a,b\}$ and $\{c\}$ are not isomorphic to each other. In $\mathbb{S}\mathbb{e}\mathbb{t}\mathbb{s}$ and many other categories, e.g. in all topoi and adhesive categories, the relation $\models\!\!\!\sim_{\langle q,m\rangle}$ for a given rule $q = L \leftarrow\!\alpha\!- K -\!\beta\!\rightarrow R$ and a match $m\colon L \to A$ is essentially right-unique provided that the left rule morphism $\alpha\colon K \rightarrowtail L$ is monic.

The other issue, namely existence of direct derivation diagrams for a given rule and match candidate, can be explained in terms of the two phenomena which are known as *deletion-preservation conflicts* and *inhibition effects*.

Example 2.10 (Deletion-preservation conflict) Let $q = \{a,b\} \leftarrowtail \{b\} \rightarrowtail \{b\}$ be a rule in $\mathbb{S}\mathbb{e}\mathbb{t}\mathbb{s}$ and let $m\colon \{a,b\} \to \{c\}$ be a match candidate. Then there does not exist any direct derivation diagram for the rule q and the match candidate m, and hence $\{c\} \models\!\!\!\sim_{\langle q,m\rangle} B$ does not hold for any set B (see Figure 19(a)).



(a) Invalid DPO-match (b) Rewriting using SPO instead

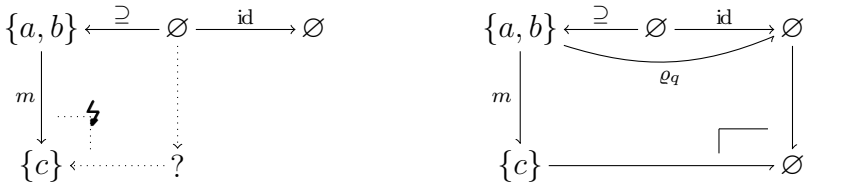
Figure 19: Deletion-preservation conflicts

The name *deletion-preservation conflict* can be explained by thinking of the rule $q = \{a,b\} \leftarrowtail \{b\} \rightarrowtail \{b\}$ as a process which *deletes* (the image of) a from the codomain of any match while *preserving* (the image of) b. However, in the example case, the element c is both the image of a and the image of b; hence the rule/morphism pair q/m does not give rise to any consistent derivation step since c would have to be deleted *and* preserved, which yields a conflict.

As a final remark on this example, in the case of SPO-rewriting, when interpreting q as a partial function $\varrho_q\colon \{a,b\} \rightharpoonup \{b\}$ and the function m as

a partial function $m\colon \{a,b\} \rightharpoonup \{c\}$, this conflict is solved by giving deletion priority, which means that ϱ_q rewrites $\{c\}$ at m to the empty set $\varnothing$, i.e. $\{c\} \rightleftharpoons_{\langle \varrho_q, m\rangle} \varnothing$ (see Figure 19(b)). Moreover, deletion-preservation conflicts cannot arise when the match candidate is monic. In fact, in the category of $\mathbb{Sets}$, for a given rule $q = L \hookleftarrow K \rightarrowtail R$ all injections $m\colon L \rightarrowtail A$ are DPO-matches. ☙

However deletion-preservation conflicts are not the only phenomenon that explains the existence of invalid DPO match candidates. In other words there are inhibition effects which are not deletion-preservation conflicts. They occur in the category of $\mathbb{Sets}$ if a match candidate for a rule identifies two elements of the left hand side that are not in the (image of) the gluing set; in this case the match candidate is said to violate the *identification condition*.



(a) Invalid DPO-match (b) Rewriting using SPO instead

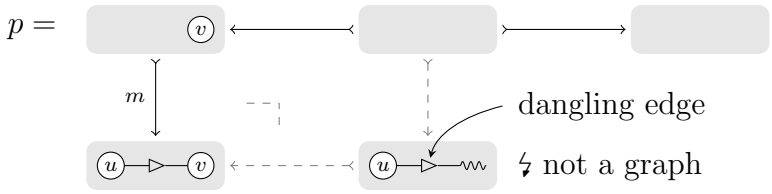
Figure 20: Identification condition

☙ Example 2.11 (Identification condition) Working in the category of $\mathbb{Sets}$, consider the rule $q = \{a,b\} \leftarrow \varnothing \rightarrow \varnothing$ and the function $m\colon \{a,b\} \rightarrow \{c\}$ as match candidate. Then the function m is not a valid DPO-match. However, using the single pushout approach in the category of partial functions $\mathbb{Pfn}$, taking m as a total partial function, the result of rewriting $\{c\}$ via the partial function $\varrho_q = \{a,b\} \rightharpoonup \varnothing$ at m is the empty set $\varnothing$. ☙

The other typical class of inhibition effects occurs in the context of the category of $\mathbb{Graphs}$ and consists of all those pairs of rules and match candidates that specify the deletion of a node while leaving an adjacent edge of this node *dangling*. This class of inhibition effects cannot be illustrated in the category of $\mathbb{Sets}$, which is related to the fact that the subset poset $\wp(A)$ for any set $A \in \mathbb{Sets}$ is a Boolean algebra, i.e. each element $X \in \wp(A)$ has a complement, namely $X^{\complement} = A \setminus X$.

☙ Example 2.12 (Dangling condition) In this example, graphs are depicted as usual: a *node* or *vertex* v is represented by a small circle, e.g. $\textcircled{v}$, and an *arc* or *edge* e from u to v is drawn as a line with an arrow head in the middle, which

indicates the orientation, e.g. $(u) \triangleright\!\!-\!\!\overset{e}{-}\!\!\triangleleft (v)$ or just $(u) \triangleright\!\!-\!\!\triangleleft (v)$. Now let $p = (v) \hookleftarrow \varnothing \hookrightarrow \varnothing$ be a rule where $\varnothing$ is the empty graph, and let $m\colon (v) \hookrightarrow (u) \triangleright\!\!-\!\!\triangleleft (v)$ be the graph morphism which embeds the node (v) into $(u) \triangleright\!\!-\!\!\triangleleft (v)$. Then m is *not* a match for the rule q; the reason for this might be illustrated as follows.

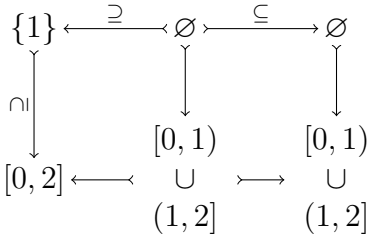


However, when interpreting p as a partial graph morphism $\varrho_p\colon (v) \rightharpoonup \varnothing$ and working in the category of graphs and partial graph morphisms, then – using the SPO approach – the rule ϱ_p rewrites $(u) \triangleright\!\!-\!\!\triangleleft (v)$ at the match m to (u), i.e. all dangling edges are removed to obtain a proper graph. ꕥ

For the final illustration of an inhibition effect that once more demonstrates the differences between the SPO and DPO approach, consider the category $\mathbb{T}\mathbb{o}\mathbb{p}$, which has topological spaces as objects, and continuous functions as morphisms. In comparison to the typical examples in the area of the algebraic approaches to rewriting [Ehrig et al., 2006], this category is less similar to the category of sets and will be used in several places of this thesis as the paradigmatic example of weakly adhesive categories, which will be introduced below in Definition 3.1.

ꕥ Example 2.13 (Topological inhibition effects) In this example all subsets of the real numbers $\mathbb{R}$ will be considered as subspaces of $\mathbb{R}$ with the usual topology. Now consider the rule $q = \{1\} \hookleftarrow \varnothing \hookrightarrow \varnothing$ in $\mathbb{T}\mathbb{o}\mathbb{p}$ and the embedding $m = \{1\} \hookrightarrow [0,2]$ as match candidate where $[0,2]$ is the subspace of $\mathbb{R}$ that is induced by the closed interval $\{x \in \mathbb{R} \mid 0 \leq x \leq 2\}$. Anticipating parts of the discussion of the relation between the DPO and SPO approach, this rule can also be interpreted as a partial continuous function $\varrho\colon \{1\} \rightharpoonup \varnothing$, and in the same way m gives rise to a match candidate in the category of topological spaces and partial continuous functions[13]. Further the result of single pushout rewriting is the space $[0,1) \cup (1,2]$ where the (half-open) intervals $[0,1)$ and $(1,2]$ denote the respective subspaces of $\mathbb{R}$.

[13]The category of topological spaces and partial continuous functions is nothing else but $\mathrm{Par}(\mathbb{T}\mathbb{o}\mathbb{p})$ as presented in Definition 2.32 below.



To see why the morphism $m\colon \{1\} \hookrightarrow [0,2]$ in $\mathbb{T}\mathbb{o}\mathbb{p}$ is not a valid DPO-match, consider the displayed double square diagram, which is suitable to describe the single pushout rewriting step $\{1\} \rightleftharpoons_{\varrho} \Rightarrow [0,1) \cup (1,2]$ in the category $\mathbb{T}\mathbb{o}\mathbb{p}$ as detailed in Section 2.5. Though $[0,1) \cup (1,2]$ might seem a good candidate to obtain a pushout square on the left because the equation $[0,2] = \{1\} \cup [0,1) \cup (1,2]$ holds for the underlying sets, the left square is not a pushout since the pushout of $\{1\}$ and $[0,1) \cup (1,2]$ over $\varnothing$ is the disjoint sum $\{1\} + ([0,1) \cup (1,2])$ in $\mathbb{T}\mathbb{o}\mathbb{p}$. In fact there does not exist any suitable pushout square (since existence of such a pushout square would contradict the fact that $[0,2]$ is connected). ⊛

These examples might suffice to give a first impression of what can be done with single and double pushout rewriting and in what sense the DPO-approach is the more restrictive one. Before elaborating on the technical background which is needed for an in depth exploration of the two approaches and their differences and similarities, the concept of concurrency and its formal rendering will be the next topic.

2.3 Concurrency as independence

Chomsky grammars are usually used to generate languages, i.e. sets of words; hence the precise relations between different ways to derive one and the same word from the start symbol are not a primary concern. The situation is different in the typical application domains of single and double pushout rewriting. The latter approaches are often used to model event driven concurrent and distributed systems by representing systems states as objects, say $A, B, C \in \mathbb{C}$, and events in the system as rewriting steps, say $A \rightleftharpoons_{\langle q,m \rangle} \Rightarrow B$ or $A \rightleftharpoons_{\langle \varrho,m \rangle} \Rightarrow B$ (depending on the chosen rewriting mechanism).

Especially if a system is large or distributed, several events might occur in unrelated subsystems, and then it is often the case that the exact order in which two events take place is accidental since – at least theoretically – the two events might actually happen at the same time, i.e. *concurrently*. The central point is that two concurrent events can occurr at the same time but could have occurred independent of each other in any order.

The latter phenomenon is a typical example of *true concurrency* where the prefix '*true*' is added to distinguishe it from other notions of concurrency, as for example interleaving semantics, which merely "assemble" all possible sequences

of events into a labelled transition system (LTS); this possibly leads to spurious dependencies between transitions, which do not have any counterpart in the modelled system. In other words, LTS semantics do not take into account the (causal) dependencies between events in the modelled system.

☼ Definition 2.14 (Labelled transition system) A *labelled transition system* is a triple $\langle S, \Lambda, \rightsquigarrow \rangle$ where S is a set of *states*, Λ is a set of *labels*, and $\rightsquigarrow \subseteq S \times \Lambda \times S$ is the *(labelled) transition relation.* ☼

There are extensions of LTSs which have an additional component that describes the dependencies between transitions; examples are the transition systems with independence (LTSI) of [Sassone et al., 1996], or refinements of these [Hildebrandt and Sassone, 1996].

However, the fully explicit representation of independence in labelled transition systems still suffers from the defect that all reachable system states must be enumerated. For example, the transition system (with independence) corresponding to the Petri net that consists of n "parallel" copies of a single enabled transition ◉→□→○ has 2^n states. The implicit partial order representation of dependence or independence of transitions like in Petri nets often allows for more compact system descriptions.

However, also in this thesis, before the topic of "compact", Petri net like system models is addressed, the starting point for the description of the fundamental concepts of concurrency in single and double pushout rewriting is the *raw transition system* (RTS) of a given set of rules $\mathscr{R}$. The states of the raw transition system are the objects of a fixed category $\mathbb{C}$ in which rewriting takes place. Further, independent of whether SPO or DPO rewriting is used, each transition between two states $A, B \in \mathbb{C}$ is labelled by a complete direct derivation diagram which witnesses the rewriting step $A \models r \Rightarrow B$ for some rule $r \in \mathscr{R}$. Then the first question will be, when and why pairs of branching or sequential transitions are independent of each other, or not.

☼ Definition 2.15 (Raw transition system) Let $\mathscr{R}$ be a set of rules in $\mathbb{C}$, and let Λ be the set[14] of all direct derivation diagrams. The *raw transition system* of $\mathscr{R}$ is the LTS $\langle \mathrm{ob}(\mathbb{C}), \Lambda, \rightsquigarrow \rangle$ where $\rightsquigarrow \subseteq \mathrm{ob}(\mathbb{C}) \times \Lambda \times \mathrm{ob}(\mathbb{C})$ is the largest relation such that for each triple $\langle A, \mathfrak{X}, B \rangle \in \rightsquigarrow$, the direct derivation diagram

[14] Theoretically, it is possible that the collection of all derivation diagrams is not a set and hence the definition does not apply. However, assuming choice and local smallness of $\mathbb{C}$, one can first obtain a set of states by restricting attention to the set of objects of (a skeletal subcategory of) $\mathbb{C}$ which is reachable from a given (set of) start state(s); in a second step, relative to the given set of rules, a set of derivation diagrams will suffice after choosing a minimal number of representatives of each isomorphism class of direct derivation diagrams.

$\mathfrak{X} \in \Lambda$ witnesses that A can be rewritten to B via some rule $r \in \mathscr{R}$. ☼

Though raw transition systems do not contain any (explicit) information concerning the dependence of pairs of branching or sequential transitions, the literature on SPO- and DPO-rewriting provides notions of parallel and sequential independence that relate the involved derivation diagrams in a suitable way (see e.g. [Rozenberg, 1997]), and thus allow to supply the missing information. The first concept is *parallel independence* which may apply to derivation diagrams which witness a pair of branching derivation steps such as $B_1 \Leftarrow_{q_1} A \Rightarrow_{q_2} B_2$; it formalizes the fact that neither of the modelled events interferes with the other one as illustrated in Example 2.17 below.

☼ Definition 2.16 (Double pushout parallel independence)
Let $q_1 = L_1 \leftarrow_{\alpha_1} K_1 \rightarrow_{\beta_1} R_1$ and $q_2 = L_2 \leftarrow_{\alpha_2} K_2 \rightarrow_{\beta_2} R_2$ be rules with monic α_1 and α_2, and let $m_1 \colon L_1 \to A$ and $m_2 \colon L_2 \to A$ be DPO-matches for q_1 and q_2, respectively. Let the left diagram in Figure 21 comprise a pair of corresponding direct derivation diagrams. These two direct derivation diagrams

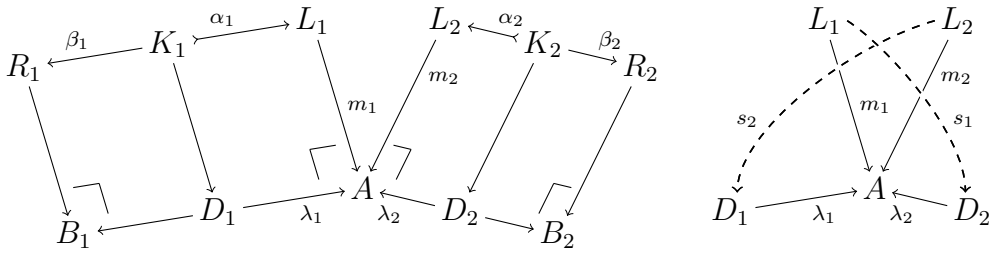
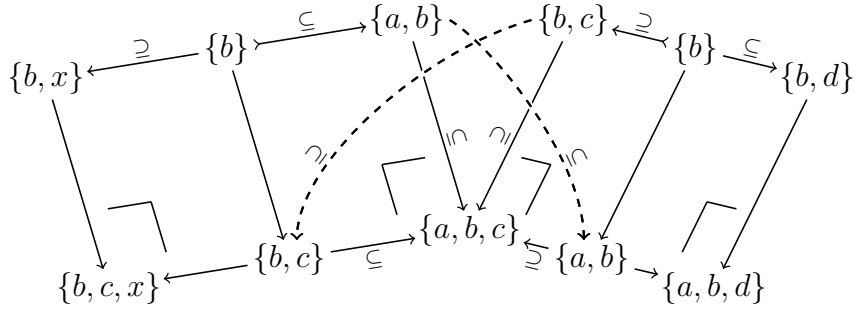


Figure 21: Parallel independence of two DPO direct derivation diagrams

are *parallel-independent* if there exists a pair of morphisms $s_2 \colon L_2 \to D_1$ and $s_1 \colon L_1 \to D_2$ that make the right hand diagram in Figure 21 commute, i.e. s_1 and s_2 must satisfy the equations $m_1 = \lambda_2 \circ s_1$ and $m_2 = \lambda_1 \circ s_2$. In this situation the pair $\langle s_1, s_2 \rangle$ is called an *independence pair* of the two direct derivation diagrams. ☼

☙ Example 2.17 (Parallel independence in $\mathbb{Sets}$) Consider the two rules $q_1 = \{a, b\} \leftarrow_{\supseteq} \{b\} \rightarrow_{\supseteq} \{b, x\}$ and $q_2 = \{b, c\} \leftarrow_{\supseteq} \{b\} \rightarrow_{\supseteq} \{b, d\}$ with the inclusions into the set $\{a, b, c\}$ as matches for these rules; then the resulting

derivation diagrams are parallel-independent.



This example also illustrates the possibility of shared "read-only" access to the element $b \in \{a, b, c\}$.

In the category of $\mathbb{S}\mathbb{e}\mathbb{t}\mathbb{s}$, a pair of direct derivation diagrams as in Figure 21 is parallel-independent if and only if the intersection of the (images of the) left hand sides is contained in the intersection of the (images of the) gluing sets, i.e. if L_1 "$\cap$" $L_2 \subseteq K_1$ "$\cap$" K_2 holds, which is shorthand for

$$m_1(L_1) \cap m_2(L_2) \subseteq m_1(\alpha_1(K_1)) \cap m_2(\alpha_2(K_2)).^{15} \tag{1}$$

This characterization can be adapted for graphs and objects of any other category that has a notion of image with suitable properties (see for example [Habel et al., 2001, Lemma 5.2]).

Further, in the category of $\mathbb{S}\mathbb{e}\mathbb{t}\mathbb{s}$, yet another characterization of parallel independence can be given: the inclusion of Equation (1) is equivalent to the equality $m_1(L_1 \setminus \alpha_1(K_1)) \cap m_2(L_2 \setminus \alpha_2(K_2)) = \varnothing$; however the latter description depends on the Boolean lattice structure of subset posets.

In the category of $\mathbb{G}\mathbb{r}\mathbb{a}\mathbb{p}\mathbb{h}\mathbb{s}$, there is a Local Church-Rosser Theorem (cf. [Habel et al., 2001, Theorem 5.3]), which states that any two derivation steps $B_1 \Leftarrow\langle q_1, m_1\rangle\Rightarrow A \Rightarrow\langle q_2, m_2\rangle\Rightarrow B_2$ which are witnessed by a pair of parallel-independent derivation diagrams as in Definition 2.16 can be completed to a diamond, i.e. there are matches $n_2 \colon L_2 \to B_1$ and $n_1 \colon L_1 \to B_2$ such that $B_1 \Rightarrow\langle q_2, n_2\rangle\Rightarrow C \Leftarrow\langle q_1, n_1\rangle\Rightarrow B_2$ holds for some graph C. The "delayed" application of q_1 at n_1 after the rewriting step $A \Rightarrow\langle q_2, m_2\rangle\Rightarrow B_2$ formalizes the fact that the second event modelled by $B_2 \Rightarrow\langle q_1, n_1\rangle\Rightarrow C$ does not causally depend on the first event, which corresponds to $A \Rightarrow\langle q_2, m_2\rangle\Rightarrow B_2$. Moreover, the two corresponding ways to sequentially execute q_1 and q_2 are sequential-independent.

[15] Given a function $f \colon A \to B$ and a subset $A' \subseteq A$, the expression $f(A') \subseteq B$ denotes the *image of* A' *under* f, i.e. $f(A') = \{b \in B \mid \exists a \in A'.\, f(a) = b\}$.

☼ Definition 2.18 (Double pushout sequential independence)
Let $q_1 = L_1 \leftarrow \alpha_1 \prec K_1 - \beta_1 \rightarrow R_1$ and $q_2 = L_2 \leftarrow \alpha_2 \prec K_2 - \beta_2 \rightarrow R_2$ be rules with monic α_1 and α_2, and let $m_1 \colon L_1 \to A$ and $n_2 \colon L_2 \to B_1$ be matches for q_1 and q_2, respectively, such that $A \Rightarrow_{\langle q_1, m_1 \rangle} B_1$ and $B_1 \Rightarrow_{\langle q_2, n_2 \rangle} C$ hold. Further let the left hand diagram in Figure 22 comprise a pair of corresponding direct derivation diagrams. This pair of direct derivation diagrams is *sequential-*

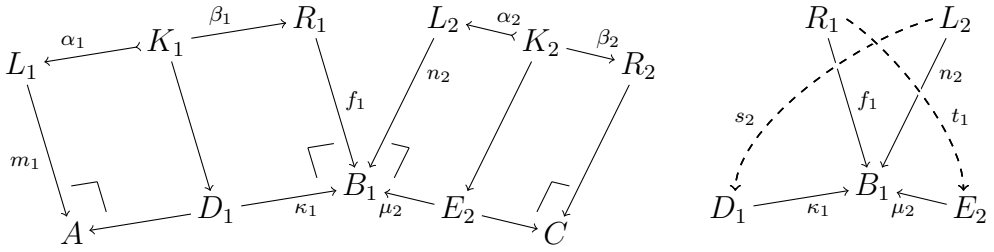


Figure 22: Sequential independence of two DPO direct derivation diagrams

independent if there exists a pair of morphisms $s_2 \colon L_2 \to D_1$ and $t_1 \colon R_1 \to E_2$ that make the right hand diagram in Figure 22 commute, i.e. the morphisms s_2 and t_1 must satisfy the equations $n_2 = \kappa_1 \circ s_2$ and $f_1 = \mu_2 \circ t_1$. In this situation the pair $\langle s_2, t_1 \rangle$ is called an *independence pair* of the two direct derivation diagrams. ☼

For sequential independence, already in the category of $\mathbb{G}\mathbb{r}\mathbb{a}\mathbb{p}\mathbb{h}\mathbb{s}$, there is no equivalent characterization in terms of images as in Equation (1). The reason for this is illustrated in the next example, which is Example 6.3 of [Habel et al., 2001]. However for the case of *linear* rules, i.e. for those rules which consist of a pair of monomorphisms, sequential independence is equivalent to the inclusion R_1 "$\cap$" $L_2 \subseteq K_1$ "$\cap$" K_2, which in the situation of Definition 2.18 is shorthand for $f_1(R_1) \cap n_2(L_2) \subseteq f_1(\beta_1(K_1)) \cap n_2(\alpha_2(K_2))$ (cf. [Habel et al., 2001, Lemma 6.2]). In this special case, the inclusion rougly corresponds to the fact that the application of the second rule does not remove anything which has not already been present before the first rule was applied. However, as mentioned before, this more intuitive account does not generalize because of the following counterexample.

❦ Example 2.19 (Sequential independence in $\mathbb{G}\mathbb{r}\mathbb{a}\mathbb{p}\mathbb{h}\mathbb{s}$)
Consider the following two rules in the category of $\mathbb{G}\mathbb{r}\mathbb{a}\mathbb{p}\mathbb{h}\mathbb{s}$: the first rule coalesces two nodes, say $q_1 = \textcircled{w}\,\textcircled{v} \leftarrow \textcircled{w}\,\textcircled{v} \rightarrow \textcircled{u}$, and the second one only checks for existence of a single node with a loop, e.g. $q_2 = \textcircled{u}\!\circlearrowleft \leftarrow \textcircled{u}\!\circlearrowleft \rightarrow \textcircled{u}\!\circlearrowleft$. As illustrated in Figure 23, the rule q_1 can be applied at the inclusion of

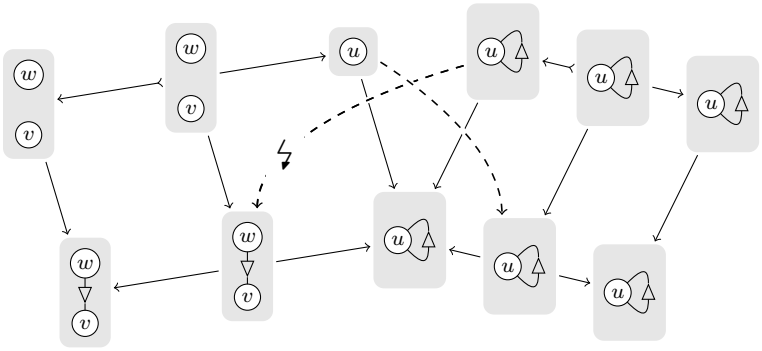


Figure 23: Failure of sequential independence of two direct derivation diagrams

its left hand side into the single edge $w \triangleright v$, which yields exactly the single node loop u; in a second step q_2 "rewrites" this loop to itself. The pair of the corresponding direct derivation diagrams is not sequential-independent, which is related to the fact that the two rules cannot be applied in the reverse order. This formalizes the fact that the event modelled by the application of the rule q_2 causally depends on the first one, which corresponds to the first transition via the rule q_1.

In fact, in the category of $\mathbb{G}\mathtt{raphs}$, local confluence for the DPO approach has the so-called *sequential commutativity* theorem as a companion [Habel et al., 2001, Theorem 6.4]. It states that if two derivation steps $A \Rightarrow_{\langle q_1, m_1 \rangle} B_1 \Rightarrow_{\langle q_2, n_2 \rangle} C$ are witnessed by a pair of sequential-independent direct derivation diagrams, then there exists another pair of sequential-independent direct derivation diagrams that witness two derivation steps of the form $A \Rightarrow_{\langle q_2, m_2 \rangle} B_2 \Rightarrow_{\langle q_1, n_1 \rangle} C$, which lead to the same object C and are obtained by switching the order in which the two rules are applied using a witnessing independence pair. However, this formulation does only give indirect information about the involved direct derivation diagrams.

Since derivation diagrams are the main object of investigation in this thesis, a more detailed account will given below in Lemma 2.30. Anticipating these details about local confluence and sequential commutativity, the illustration in Figure 24 might nevertheless convey enough information concerning the interplay of independence pairs for parallel and sequential independence.

Suitable notions of parallel and sequential independence for derivation diagrams with the corresponding confluence and commutativity results are an essential ingredient and the starting point of investigations into the concurrent semantics of double pushout rewriting. Here, concurrent semantics is under-

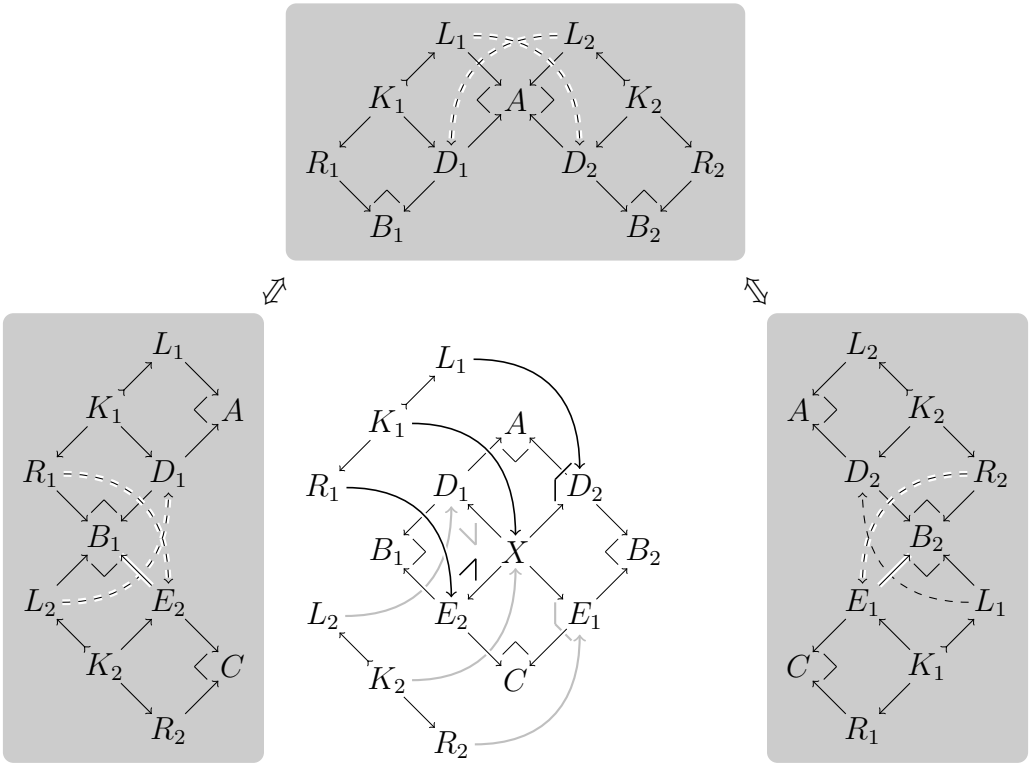


Figure 24: The diamond of local confluence and sequential commutativity

stood as in [Baldan, 2000], which is the paradigmatic treatise for the special case of the category of Graphs. One of the characteristics of the latter work is an extensive account of the connection between graph transformation on the one hand and the theory of Petri nets on the other hand with the focus on unfoldings.

One of the main challenges of the present thesis is to find a conceptually adequate characterization of a wide class of categories that allows to lift the work on unfoldings of [Baldan, 2000] from the concrete category of Graphs to the general *abstract* level. An example of a similar project is the generalization of standard theorems of DPO graph transformation to DPO rewriting in weak adhesive HLR categories, as developed in [Ehrig and Prange, 2006, Ehrig et al., 2006]. The starting point of the latter work and also the present thesis are adhesive categories.

2.4 Adhesive categories

Adhesive categories are generalizations of the categories of $\mathbb{S}\mathrm{ets}$ and $\mathbb{G}\mathrm{raphs}$, in other words both $\mathbb{S}\mathrm{ets}$ and $\mathbb{G}\mathrm{raphs}$ are examples of adhesive categories; this class of categories has been introduced in [Lack and Sobociński, 2005]. Their main defining property is the validity of a (general) Van Kampen theorem (see [Brown and Janelidze, 1997, Definition 1.1]); the following explanation of adhesivity is given in [Lack and Sobociński, 2005, p. 5–6].

> The definition of adhesive category is stated in terms of something called a *Van Kampen square*, which can be thought of as a "well-behaved pushout"[...]. The name Van Kampen derives from the relationship between these squares and the Van Kampen theorem in topology, in its "coverings version"[...].

Among the different, equivalent characterizations[16] of Van Kampen squares, the following one uses only a minimal repertoire of notions from category theory.

☼ Definition 2.20 (Van Kampen square) Let $B \leftarrow f - A - m \rightarrow C$ be a span, and let $B - n \rightarrow D \leftarrow g - C$ be its pushout.

The resulting pushout square is *Van Kampen* (VK), if for each commutative cube as in Figure 25 on the left, having pullback squares as back faces, its top face is a pushout square if and only if its front faces are pullback squares. ☼

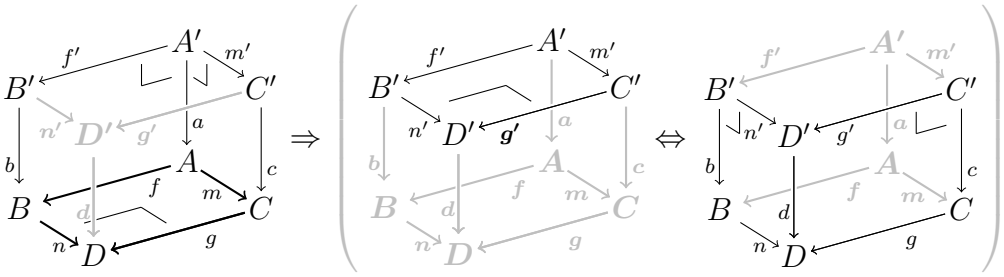


Figure 25: Van Kampen square

The discussion of the basic properties of Van Kampen squares listed in [Lack and Sobociński, 2005] is deferred to Section 7. Examples of Van Kampen squares include all those pushout squares in $\mathbb{S}\mathrm{ets}$ (or in any other

[16] See [Lack and Sobociński, 2005, Proposition 2.5] for a list of equivalent properties.

topos $\mathbb{E}$) which arise by taking the pushout of a span having at least one monomorphism amongst its two arrows. The definition of adhesive categories, abstracting away from the special cases of $\mathbb{S}\mathbb{e}\mathbb{t}\mathbb{s}$, $\mathbb{G}\mathbb{r}\mathbb{a}\mathbb{p}\mathbb{h}\mathbb{s}$ and topoi, is as follows.

Definition 2.21 (Adhesive categories) A category $\mathbb{C}$ is *adhesive* if

- it has pullbacks,
- it has pushouts along monomorphisms, i.e. pushouts of diagrams of the form $B \leftarrow f - A \rightarrowtail m \rightarrow C$ with monic m exist, and
- pushouts along monomorphisms yield Van Kampen squares.

Among the numerous properties that adhesive categories share with the category of $\mathbb{S}\mathbb{e}\mathbb{t}\mathbb{s}$ (see [Lack and Sobociński, 2005, Lack and Sobociński, 2006]), uniqueness of pushout complements is the one which makes (the most widely used variant of) double-pushout rewriting essentially deterministic.

Fact 2.22 (Unique pushout complements ([Lack and Sobociński, 2005])) Let $\mathbb{C}$ be an adhesive category, let $m\colon A \rightarrowtail C$ be a monomorphism, and let $g\colon C \to D$ be any morphism. Further let $f\colon A \to B$ and $n\colon B \to D$, and $f'\colon A \to B'$ and $n'\colon B' \to D$ be two *pushout complements* for $C \leftarrow g - D \leftarrow m \prec A$, i.e. pairs of composable arrows such that $B - n \rightarrow D \leftarrow g - C$ and $B' - n' \rightarrow D \leftarrow g - C$ are pushouts of $B \leftarrow f - A \rightarrowtail m \rightarrow C$ and $B' \leftarrow f' - A \rightarrowtail m \rightarrow C$, respectively (see the premise in Figure 26).

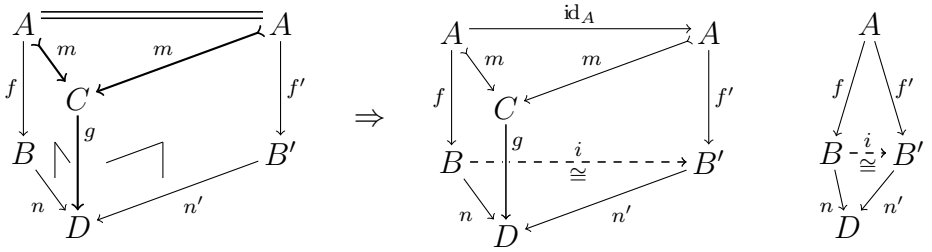


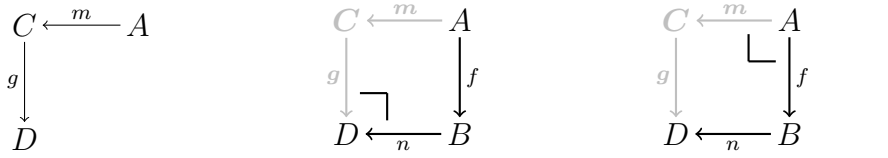
Figure 26: Uniqueness of pushout complements

Then there exists a unique isomorphism $i\colon B \to B'$ for which both $f' = i \circ f$ and $n' \circ i = n$ hold (see the conclusion in Figure 26).

Corollary 2.23 (Essential uniqueness of DPO-rewriting) Let $\mathbb{C}$ be any adhesive category, and let $q = L \leftarrow \alpha \prec K - \beta \rightarrow R$ be a *left-linear* rule, i.e. one with monic α, and let $m\colon L \to A$ be an (arbitrary) match.

Then the result of rewriting A at m via q is essentially unique, which means that any two objects $B, B' \in \mathbb{C}$ that satisfy $A \Rightarrow\!\!\!\!\!\!\cdot\;\langle q,m\rangle\Rightarrow B$ and $A \Rightarrow\!\!\!\!\!\!\cdot\;\langle q,m\rangle\Rightarrow B'$ are isomorphic, i.e. $B \cong B'$.

Extrapolating from the examples in the category of $\mathbb{G}\mathtt{raphs}$, the operation of taking pushout complements can be thought of as the *deletion mechanism* of double pushout rewriting. However, since pushout complements do not exist in general, pushout complementation is only a partial operation (cf. Example 2.11 and Example 2.12 concerning the inhibition condition and dangling edges, respectively). To obtain an unconditional deletion mechanism, pushout complementation will be replaced by *pullback* complementation; as a "side effect", as described below in Fact 2.43, using (a certain class of) pullback complements instead of pushout complements, transforms double pushout rewriting to single pushout rewriting.



(a) Composable pair (b) Pushout complement (c) Pullback complement

Figure 27: Pushout and pullback complements

☼ Definition 2.24 (Pushout and pullback complements) Let $\mathbb{C}$ be any category, let $m\colon A \to C$ and $g\colon C \to D$ be a pair of composable morphisms, i.e. a pair of morphisms such that their composite $g \circ m$ is defined.

A pair of composable morphisms $f\colon A \to B$ and $n\colon B \to D$ is a *pushout complement (pullback complement)*[17] if the resulting square is a pushout square (pullback square), i.e. if $C \mathrel{-g\!\!\rightarrow} D \mathrel{\leftarrow\!\!n-} B$ is a pushout of $C \mathrel{\leftarrow\!\!m-} A \mathrel{-f\!\!\rightarrow} B$ (the span $C \mathrel{\leftarrow\!\!m-} A \mathrel{-f\!\!\rightarrow} B$ is a pullback of $C \mathrel{-g\!\!\rightarrow} D \mathrel{\leftarrow\!\!n-} B$). ☼

Uniqueness of pushout complements for composable pairs $D \mathrel{\leftarrow\!\!g-} C \mathrel{\leftarrow\!\!m\prec} A$ with monic m is one of the reasons why the major part of this thesis will be concerned with left-linear rules. Further, left-linear rules are also relevant to the standard comparison between single and double pushout rewriting in [Ehrig et al., 1997]. More detailed, the latter SPO/DPO comparison depends

[17] The pullback complements of [Dyckhoff and Tholen, 1987] will be referred to as *final* pullback complements in this thesis.

on the fact that pushout complements as in Fact 26 are also pullback complements of a particular kind, which made more precise below (see Fact 2.33).

☼ Definition 2.25 (Left-linear rule) Let $\mathbb{C}$ be any category and further let $q = L \leftarrow\!\alpha\!- K -\!\beta\!\rightarrow R$ be a rule in $\mathbb{C}$. Then $\mathbb{C}$ is *left-linear* if the left morphism $L \leftarrow\!\alpha\!\prec K$ is monic. ☼

Before coming back to concurrency as independence in the setting of adhesive categories, consider the following fact as an example of the "good" behaviour of pushouts along monomorphisms. A variant of it will recur as one of the defining conditions of weak adhesive HLR categories (see Definition 2.45).

◎ Fact 2.26 (Pushouts of monomorphisms) In an adhesive category, monomorphisms are stable under pushout and pushouts along monomorphisms give rise to pullback squares.

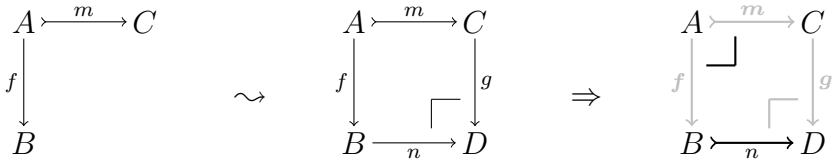


Figure 28: Pushouts along monomorphisms in adhesive categories

As illustrated in Figure 28, this means that given a pushout $B -\!n\!\rightarrow D \leftarrow\!g\!- C$ of a span $B \leftarrow\!f\!- A \rightarrowtail\!m\!\rightarrow C$ with monic m, the morphism n is monic, and moreover the span $B \leftarrow\!f\!- A \rightarrowtail\!m\!\rightarrow C$ is a pullback of $B -\!n\!\rightarrow D \leftarrow\!g\!- C$. This implies that in any pushout complement $D \leftarrow\!n\!- B \leftarrow\!f\!- A$ of a given composable pair $D \leftarrow\!g\!- C \leftarrow\!m\!\prec A$ with monic m, the morphism n is monic (see also Figure 27).

As mentioned before, adhesivity of a category $\mathbb{C}$ ensures local confluence and sequential commutativity for double pushout rewriting in $\mathbb{C}$. However, throughout large portions of this thesis, attention will be further restricted to that variant of double pushout rewriting which has been shown to be most expressive in Section 4.1 of [Habel et al., 2001], in the category of $\mathbb{G}\mathtt{raphs}$. This version of DPO-rewriting uses only left-linear rules and monic matches. As a minor complication, the notion of sequential independence has to be adapted to obtain sequential commutativity.

☼ Definition 2.27 (Left-linear, monic double pushout rewriting) Let $q = L \leftarrow\!\alpha\!\prec K -\!\beta\!\rightarrow R$ be a left-linear rule, and let $m\colon L \rightarrowtail A$ in $\mathbb{C}$ be a monomorphism; in this situation m is a DPO_{llm} *match candidate* or *proto-match* for q. Further, a DPO_{llm} *direct derivation diagram for* q *and* m is nothing else but a DPO

direct derivation diagram for q and m, and m is a DPO_{llm} *match for* the rule q if m is a DPO match for q.

The rewriting mechanism of DPO_{llm} is exactly the DPO-approach as given in Definition 2.7, and hence no new notation is introduced concerning the rewriting relation. In contrast, the notion of sequential independence is not appropriate any more as illustrated in the next example, which is Example 6.5 of [Habel et al., 2001].

Example 2.28 (Failure of sequential commutativity) Consider the following identity rule on an edge $q_1 = \textcircled{v}\triangleright\textcircled{w} \leftarrowtail \textcircled{v}\triangleright\textcircled{w} \rightarrow \textcircled{v}\triangleright\textcircled{w}$ and the fusion of a pair of nodes $q_2 = \textcircled{v}\ \textcircled{w} \leftarrowtail \textcircled{v}\ \textcircled{w} \rightarrow \textcircled{u}$; these rules can be applied sequentially at the embeddings into the one edge graph $\textcircled{v}\triangleright\textcircled{w}$. The result of rewriting is the single loop $\textcircled{u}$ as illustrated in Figure 29. The two displayed direct

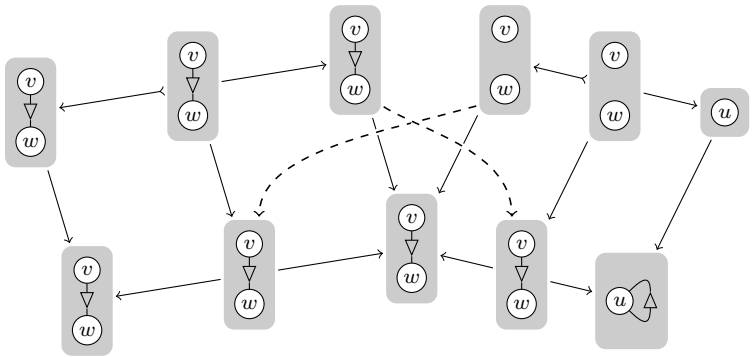


Figure 29: Counter-example for sequential commutativity for DPO_{llm}

derivations are sequential-independent. However, using the DPO_{llm}-approach to rewriting, there is no way to apply q_1 after q_2 since there is not even a match candidate from the left hand side $\textcircled{v}\triangleright\textcircled{w}$ into the single loop $\textcircled{u}$ as the only morphism between these two graphs is not monic.

This failure of sequential commutativity for pairs of sequential-independent double pushout direct derivation diagrams with *monic* matches can be mended by a stronger notion of independence, namely *strong* sequential independence. The notion of parallel independence can be strengthened in an analogous way.

Definition 2.29 (Strong sequential independence)
Let $q_1 = L_1 \leftarrow\!\alpha_1\!\prec K_1 -\!\beta_1\!\rightarrow R_1$ and $q_2 = L_2 \leftarrow\!\alpha_2\!\prec K_2 -\!\beta_2\!\rightarrow R_2$ be a pair of left-linear rules and let $m_1 \colon L_1 \rightarrowtail A$ and $n_2 \colon L_2 \rightarrowtail B_1$ be monomorphisms; further let the left one of the diagrams in Figure 30 comprise a pair of DPO_{llm} direct

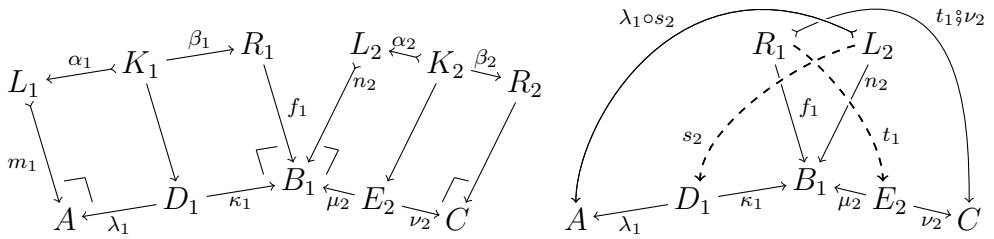


Figure 30: Strong sequential independence

derivation diagrams, which witness the two rewriting steps $A \Rightarrow_{\langle q_1, m_1 \rangle} B_1$ and $B_1 \Rightarrow_{\langle q_2, n_2 \rangle} C$.

Then these two direction derivation diagrams are *strongly sequential-independent* if there exists an independence pair $\langle s_2, t_1 \rangle$ as displayed in the right hand diagram in Figure 30, i.e. s_2 and t_1 additionally satisfy the requirement that the two morphisms $\lambda_1 \circ s_2 \colon L_2 \rightarrowtail A$ and $t_1 \fatsemi \nu_2 \colon R_1 \rightarrowtail C$ are monic; in this situation also the independence pair $\langle s_2, t_1 \rangle$ is called *strong*.

Using this refined notion of sequential independence, sequential commutativity can be recovered as a direct consequence of the next lemma, which provides some additional information about the involved (direct derivation) diagrams.

Lemma 2.30 (Sequential independence diamond) Let $\mathbb{C}$ be an adhesive category, and let $q_1 = L_1 \leftarrow \alpha_1 \prec K_1 - \beta_1 \rightarrow R_1$ and $q_2 = L_2 \leftarrow \alpha_2 \prec K_2 - \beta_2 \rightarrow R_2$ be a pair of left-linear rules; further let $m_1 \colon L_1 \rightarrowtail A$ and $n_2 \colon L_2 \rightarrowtail B_1$ be monomorphisms. Finally let the left one of the two diagrams in Figure 31 comprise a pair of DPO direct derivation diagrams with a strong independence pair $\langle s_2, t_1 \rangle$. Then this diagram can be extended to one as displayed on the right hand side in Figure 31 in which all squares are pushout squares.

The left diagram in Figure 31 witnesses the two sequential rewriting steps $A \Rightarrow_{\langle q_1, m_1 \rangle} B_1$ and $B_1 \Rightarrow_{\langle q_2, n_2 \rangle} C$. Combining the pushout squares of the right hand diagram in Figure 31 in a suitable way using the Pushout Lemma (Lemma A.14) yields two new direct derivation diagrams with a strong independence pair $\langle s_1, t_2 \rangle$ that give rise to the application of q_1 and q_2 in reverse order, corresponding to sequential rewriting steps $A \Rightarrow_{q_2} B_2$ and $B_2 \Rightarrow_{q_1} C$. In other words Lemma 2.30 has the following corollary.

Corollary 2.31 (Sequential commutativity for strong independence) Let $\mathbb{C}$ be an adhesive category. Further let $q_1 = L_1 \leftarrow \alpha_1 \prec K_1 - \beta_1 \rightarrow R_1$ and $q_2 = L_2 \leftarrow \alpha_2 \prec K_2 - \beta_2 \rightarrow R_2$ be a pair of left-linear rules, and let $m_1 \colon L_1 \rightarrowtail A$ and

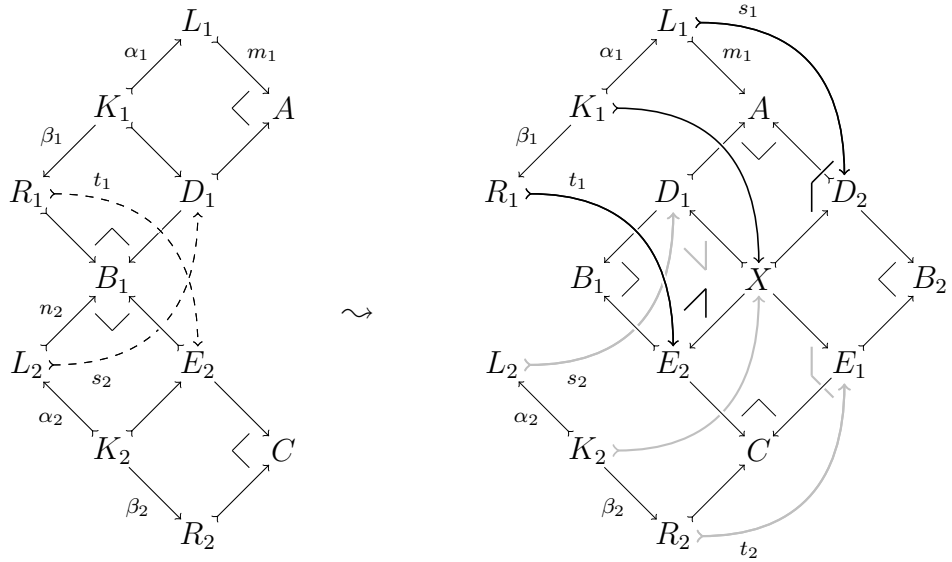


Figure 31: Sequential commutativity diamond using strong independence

$n_2 \colon L_2 \rightarrowtail B_1$ be monomorphisms; finally let the left one of the two diagrams in Figure 31 comprise a pair of strongly sequential-independent DPO direct derivation diagrams witnessing the two rewriting steps $A \Rightarrow_{\langle q_1, m_1 \rangle} B_1 \Rightarrow_{\langle q_2, n_2 \rangle} C$.

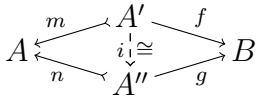
Then there exist strongly sequential-independent DPO direct derivation diagrams that witness the two rewriting steps $A \Rightarrow_{q_2} B_2$ and $B_2 \Rightarrow_{q_1} C$ for some object $B_2 \in \mathbb{C}$.

The properties of adhesive categories covered so far mainly concern facts about double pushout rewriting. However, though adhesivity is related to the DPO approach as described in the results of [Lack and Sobociński, 2005], in the context of this thesis, it also plays another role which is related to single pushout rewriting. Indeed, when rules are restricted to pairs of monomorphisms, adhesivity allows to formalize the DPO approach as the reversible variant of the SPO approach working in the partial map category and using total morphisms as matches. This latter relation between single and double pushout rewriting will be the guiding idea behind the elaborations about partial maps in adhesive categories that will follow.

2.5 Relating single- and double-pushout rewriting

To compare single and double pushout rewriting, the authors of the handbook chapter [Ehrig et al., 1997] restrict DPO-rewriting to left-linear rules, i.e. to those $\mathbb{C}$-spans $L \leftarrowtail K \to R$ in which the left morphism is monic, and consider SPO-rewriting in the category of partial maps.[18] Apart from the fact that this condition on rules ensures an essentially "deterministic" rewriting relation using DPO-rewriting (see Corollary 2.23) these spans also describe partial maps and hence can be used as rules for single pushout rewriting in the partial map category $\mathrm{Par}(\mathbb{C})$, which can be defined as follows.

☼ Definition 2.32 (Partial map category)



Let $\mathbb{C}$ be a category and let $A, B \in \mathbb{C}$ be objects. A *partial map span* from A to B is defined as a $\mathbb{C}$-span $A \leftarrow m\!\prec A' \to f\!\rightarrow B$ with monic m; two partial map spans $A \leftarrow m\!\prec A' \to f\!\rightarrow B$ and $A \leftarrow n\!\prec A'' \to g\!\rightarrow B$ are *isomorphic*[19] if there exists an isomorphism $i\colon A' \xrightarrow{\cong} A''$ such that $m = n \circ i$. As *'being isomorphic'* is an equivalence relation, one can form the isomorphism class of a partial map span $A \leftarrow m\!\prec A' \to f\!\rightarrow B$, which then is denoted by $[m_{(A')} f\rangle$ or just $[m,f\rangle$ if A' is not relevant. A *partial map* from A to B is such an isomorphism class of partial map spans from A to B as illustrated in the displayed diagram.

If $\mathbb{C}$ has pullbacks along monomorphisms, i.e. pullbacks of diagrams of the form $C \to f\!\rightarrow D \leftarrow n\!\prec B$ with monic n exist in $\mathbb{C}$, the *category of partial $\mathbb{C}$-maps* $\mathrm{Par}(\mathbb{C})$ is defined as follows: it shares the same collection of objects with $\mathbb{C}$, i.e. $\mathrm{ob}(\mathbb{C}) = \mathrm{ob}(\mathrm{Par}(\mathbb{C}))$, and the morphisms with domain $A \in \mathbb{C}$ and codomain $B \in \mathbb{C}$ are all partial maps from A to B, i.e.

$$\mathrm{Par}(\mathbb{C})(A,B) = \left\{ [m,f\rangle\colon A \rightharpoonup B \;\middle|\; \begin{array}{l} A \leftarrow m\!\prec A' \to f\!\rightarrow B \text{ is} \\ \text{a } \mathbb{C}\text{-span with monic } m \end{array} \right\}.^{20}$$

The identity on $A \in \mathbb{C}$ is given by $[\mathrm{id}_A,\mathrm{id}_A\rangle\colon A \rightharpoonup A$ in $\mathrm{Par}(\mathbb{C})$; further, to compose two partial maps $[m_{(A')} f\rangle\colon A \rightharpoonup B$ and $[n_{(B')} g\rangle\colon B \rightharpoonup C$ take any pullback $A' \leftarrow l\!\prec A'' \to h\!\rightarrow B'$ of $A' \to f\!\rightarrow B \leftarrow n\!\prec B'$ to obtain the composite $[n_{(B')} g\rangle \circ [m_{(A')} f\rangle = [m \circ l_{(A'')} g \circ h\rangle\colon A \rightharpoonup C$ (see also Figure 32).

[18] Actually, in [Ehrig et al., 1997] only the category of hyper-graphs is considered.

[19] This terminology is in accordance with the bi-categorical structure of partial map categories.

[20] As it will often be the case that both $\mathbb{C}$-morphisms and partial $\mathbb{C}$-maps occur in the same diagram, the former are depicted using normal arrow heads ($\to$) whereas for the latter partial tips ($\rightharpoonup$) will be used as in $[m,f\rangle\colon A \rightharpoonup B$.

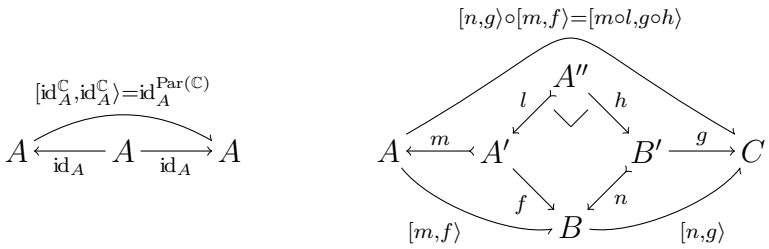


Figure 32: Identities and composition in partial map categories

A partial map $[m,f\rangle\colon A \rightharpoonup B$ in $\mathrm{Par}(\mathbb{C})$ is *total* if m is an isomorphism. Each $\mathbb{C}$-morphism $f\colon A \to B$ induces the total map $[\mathrm{id}_A,f\rangle\colon A \rightharpoonup B$ in $\mathrm{Par}(\mathbb{C})$, which is also written as $[f\rangle\colon A \rightharpoonup B$.

As adhesive categories have all pullbacks by definition, their associated category of partial maps is available. The following fact links single and double pushout rewriting in any adhesive category $\mathbb{C}$ or, more precisely, it relates SPO in $\mathrm{Par}(\mathbb{C})$ using total matches on the one hand with DPO in $\mathbb{C}$ with left-linear rules on the other hand. The main property of adhesive categories which is used to proof this fact is discussed in Section 2.5.2.

Fact 2.33 (Double pushout as a special case of single pushout) Let $\mathbb{C}$ be an adhesive category, let $q = L \leftarrow\!\alpha\!\prec K \prec\!\beta\!\rightarrow R$ be a left-linear $\mathbb{C}$-rule, and let $m\colon L \to A$ in $\mathbb{C}$ be a match which gives rise to a DPO-derivation step $A \Rightarrow_{\langle q,m\rangle} B$.

Then there is an SPO-derivation step from A to B which uses the rule $[\alpha,\beta\rangle\colon L \rightharpoonup R$ at the total match $[m\rangle\colon L \rightharpoonup A$, i.e. $A \Rightarrow_{\langle[\alpha,\beta\rangle,[m\rangle\rangle} B$.

As shown before in Examples 2.10 and 2.12, the converse of this fact does not hold – not even in the category of $\mathbb{Sets}$. However, in the category $\mathbb{Sets}$, the converse holds if not only rules are restricted to left-linear ones, but also matches are required to be monic. Indeed, the latter two conditions, namely left-linearity and monic matches, are imposed throughout the major part of this thesis.

Fact 2.34 (Double pushout as single pushout in $\mathbb{Sets}$[21]) In the category of $\mathbb{Sets}$, any SPO derivation with a monic match is a DPO-derivation, i.e. for any partial function $\varrho\colon L \rightharpoonup R$ with domain of definition[22] $L' \subseteq L$, and for any

[21] This fact holds true in any Boolean topos.

[22] Given a partial function $\varrho\colon L \rightharpoonup R$, its *domain of definition* is given by the subset $\mathrm{df}(L) = \{l \in L \mid \varrho(l) \in R\}$ and $\varrho{\restriction_{\mathrm{df}(L)}}\colon \mathrm{df}(L) \to R$ is the total function which maps each $l \in \mathrm{df}(L)$ to $\varrho(l)$.

injection $m\colon L \rightarrowtail A$ the following holds: if $A \models\!\!\!\Rightarrow_{\langle \varrho, [m\rangle\rangle} B$ is an SPO-rewriting step then $A \models\!\!\!\Rightarrow_{\langle q_\varrho, m\rangle} B$ is a DPO-rewriting step where $q_\varrho = L \leftarrow\!\supseteq\!\prec L' -\varrho\restriction_{L'}\!\rightarrow R$.

With the assumptions of this fact, the resulting set of the rewriting step $A \models\!\!\!\Rightarrow_{\langle q_\varrho, m\rangle} B$ is $B = A \setminus (L \setminus L') \cup (R \setminus L')$, provided that $m\colon L \rightarrowtail A$ is an inclusion and additionally $L' \subseteq A \cap R$ holds true (cf. the description of DPO-rewriting given in Example 2.8). A result similar to Fact 2.34 holds for the category of $\mathbb{G}\mathtt{raphs}$ if the left morphisms of rules are bijective on nodes, i.e. a graph rule $L \leftarrow\!\alpha\!\prec K -\beta\!\rightarrow R$ with $\alpha = \langle \alpha_V, \alpha_E\rangle\colon K \to L$ is only allowed if the function $\alpha_V\colon K \to L$ is a bijection.

Before delving deeper into the category theoretical background needed for a more thorough explanation of the listed similarities and differences between single and double rewriting, the main difference of the two approaches concerns their deletion mechanism. The following definition specifies the situations in which the SPO-approach gives priority to deletion over preservation to resolve deletion-preservation conflicts and violations of the identification condition, and presents the (usually wider) notion of inhibition effect, which is inherently linked to the DPO-approach.

☼ Definition 2.35 (Conflicts and inhibition effects) Let $\mathbb{C}$ be a category with pullbacks along monomorphisms; moreover let $q = L \leftarrow\!\alpha\!\prec K -\beta\!\rightarrow R$ be a left-linear rule, let $m\colon L \to A$ in $\mathbb{C}$ be a morphism, and finally let $A -g\!\rightharpoonup B \leftharpoonup\!f- R$ be a pushout of $A \leftharpoonup\![m\rangle\!- L -\![\alpha,\beta\rangle\!\rightharpoonup R$ in $\mathrm{Par}(\mathbb{C})$ (in this situation $A \models\!\!\!\Rightarrow_{\langle [\alpha,\beta\rangle, [m\rangle\rangle} B$ is an SPO rewriting step).

Then the pair q, m has a *conflict* if $f\colon R \rightharpoonup B$ is not a total morphism in $\mathrm{Par}(\mathbb{C})$. Further an *inhibition effect* occurs for the pair q, m if there does not exist any pushout complement for the composable pair $A \leftarrow\!m\!- L \leftarrow\!\alpha\!\prec K$ in the category $\mathbb{C}$. ☼

The definition and terminology are taylored for the standard examples of graph rewriting. First, the category of partial graph morhpisms has enough pushouts. Moreover, conflicts arise if the match m maps two edges e_1, e_2 in the left hand side of a rule to a single edge $e = m(e_1) = m(e_2)$ while only e_1 is in the image of the gluing object, i.e. $e_2 \notin \alpha(K) \ni e_1$; in intuitive terms, there is now a conflict between deletion and preservation of the edge e. Finally, any other situation in which there is a rewriting step according to single pushout rewriting but no double pushout step yields an inhibition effect; the typical example is the dangling edge (see Example 2.12).

In a wide range of categories, conflicts can be avoided by allowing only cer-

tain (mono-)morphisms as matches.[23] In the case of $\mathbb{G}\mathtt{raphs}$, such "restrictions" may even increase the expressive power of double pushout rewriting, a fact that has been discussed in detail in [Habel et al., 2001]. In contrast, inhibition effects are understood only in very few special cases [Llabrés and Rosselló, 1998].

2.5.1 The role of partial map classifiers

As a first step towards an answer to the question of when a DPO-step in some category $\mathbb{C}$ gives rise to an SPO-step in the associated category of partial maps $\mathrm{Par}(\mathbb{C})$, the conditions that ensure that a pushout in $\mathbb{C}$ "remains" a pushout if it is "embedded" into $\mathrm{Par}(\mathbb{C})$ will be a discussed. Equivalently, one could only consider double pushout direct derivation diagrams based on so-called *non-consuming* rules, which are $\mathbb{C}$-spans of the form $L \leftarrow_{\mathrm{id}} L \rightarrow_{\beta} R$, i.e. $\mathbb{C}$-spans that correspond to total maps in $\mathrm{Par}(\mathbb{C})$, and ask whether these also witness a corresponding SPO-rewriting step; the two questions are equivalent because a DPO direct derivation diagram for such a non-consuming rule $L \leftarrow_{\mathrm{id}} L \rightarrow_{\beta} R$ is essentially the same as a pushout along $\beta\colon L \to R$.

What exactly is meant by *embedding* a $\mathbb{C}$-pushout into the partial map category $\mathrm{Par}(\mathbb{C})$ and in what sense $\mathbb{C}$-pushout *remains* a pushout is captured by the following definition.

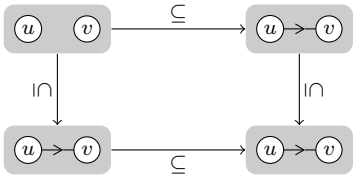
☼ Definition 2.36 (Embedding and preservation of pushout squares) Let $\mathbb{C}$ be a category with pullback along monomorphisms. Further let $C \leftarrow_g A \rightarrow_f B$ in $\mathbb{C}$ be a span, and let $C \rightarrow_h D \leftarrow_k B$ be its pushout in $\mathbb{C}$, resulting in the pushout square $\begin{smallmatrix} A & \to & B \\ C & \to & D \end{smallmatrix}$.

The *embedding* of the span and its pushout into the partial map category $\mathrm{Par}(\mathbb{C})$ is the span $C \leftarrow_{[g\rangle} A \rightarrow_{[f\rangle} B$ and the cospan $C \rightarrow_{[h\rangle} D \leftarrow_{[k\rangle} B$ in $\mathrm{Par}(\mathbb{C})$, respectively. The pushout cospan $C \rightarrow_h D \leftarrow_k B$ in $\mathbb{C}$ *remains* a pushout or is *preserved* by the embedding if $C \rightarrow_{[h\rangle} D \leftarrow_{[k\rangle} B$ is still a pushout of $C \leftarrow_{[g\rangle} A \rightarrow_{[f\rangle} B$ in $\mathrm{Par}(\mathbb{C})$. ☼

All pushouts in the categories of $\mathbb{S}\mathtt{ets}$ and $\mathbb{G}\mathtt{raphs}$ are preserved by the embedding into the respective partial map categories. This is related to the existence of partial map classifiers as defined and explained below. However not all pushouts are "robust" in this sense as illustrated by the following example.

☙ Example 2.37 (Fragile pushouts)

[23] See [Heckel and Wagner, 1995] for the concrete case of graphs; the category theoretical considerations of the present thesis are inspired by [Dyckhoff and Tholen, 1987] (see also [Corradini et al., 2006]).



Consider the displayed pushout in the category $\mathtt{sGraphs}$, which has simple graphs as objects where a *simple* graph is a pair $G = \langle V, E\rangle$ such that $E \subseteq V \times V$ is a binary endorelation; further, morphisms are relation preserving functions, which means that a morphism between two simple graph $\langle V, E\rangle$ and $\langle W, F\rangle$ is a function $f\colon V \to W$ such that $\langle f(u), f(v)\rangle \in F$ holds for all *edges* $\langle u, v\rangle \in E$. An *edge* $\langle u, v\rangle \in E$ between two nodes $\textcircled{u}$ and $\textcircled{v}$ is depicted as $\textcircled{u}\!\!\rightarrow\!\!\textcircled{v}$. Equivalently, $\mathtt{sGraphs}$ can be described as the full subcategory of $\mathtt{Graphs}$ which has all graphs without parallel edges as the collection of objects.

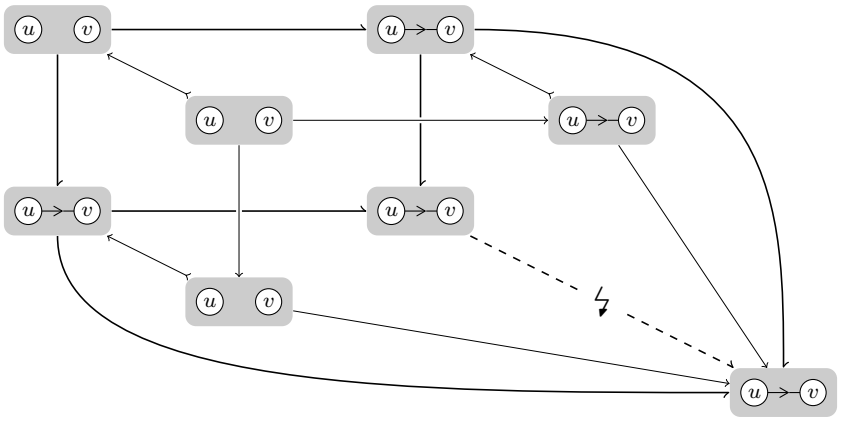


Figure 33: A pushout in $\mathtt{sGraphs}$ that is not a pushout in Par($\mathtt{sGraphs}$)

To see in what sense this pushout is "fragile", consider the same square in the partial map category Par($\mathtt{sGraphs}$), which has the same collection of objects as $\mathtt{sGraphs}$ but relation preserving *partial* functions[24] as arrows; now observe that there are more ways to close the span $\textcircled{u}\!\!\rightarrow\!\!\textcircled{v} \hookleftarrow \textcircled{u}\ \ \textcircled{v} \rightarrowtail \textcircled{u}\!\!\rightarrow\!\!\textcircled{v}$ to a square than there are in $\mathtt{sGraphs}$. One such "new" possibility is illustrated in Figure 33, which is chosen in such a way that there does not exist any mediating partial map, which means that the square is not a pushout in Par($\mathtt{sGraphs}$). ⌀

[24]This means that in Par($\mathtt{sGraphs}$), a morphism between two endorelations $E \subseteq V \times V$ and $F \subseteq W \times W$ is a partial function $f\colon V \rightharpoonup W$ such that

$$f(E) = \{\langle f(u), f(v)\rangle \mid \langle u, v\rangle \in E \text{ and } u, v \in \mathrm{df}(f)\} \subseteq F$$

where $\mathrm{df}(f) = \{v \in V \mid f \text{ is defined at } v\}$ is the domain of definition of f.

A more succinct way to state that $\mathbb{C}$-pushouts are preserved by the embedding into $\mathrm{Par}(\mathbb{C})$ is based on the fact that pushouts are colimits and that the described embedding of $\mathbb{C}$ into $\mathrm{Par}(\mathbb{C})$ extends to a functor of $\mathbb{C}$ into $\mathrm{Par}(\mathbb{C})$. Then the preservation of pushouts by the embedding is just preservation of this particular type of colimit by this functor.

☼ Definition 2.38 (Graphing functor[25]) Let $\mathbb{C}$ be a category with pullbacks (along monomorphisms), and let $\mathrm{Par}(\mathbb{C})$ the associated partial map category.

Then the *graphing functor* $\Gamma\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$ is the identity on objects and maps each $\mathbb{C}$-morphism to its functional relation, i.e. $\Gamma(C) = C$ for all $C \in \mathbb{C}$ and $\Gamma(f\colon C \to D) = [\mathrm{id}_C, f\rangle\colon C \rightharpoonup D$ for all morphisms $f\colon C \to D$ in $\mathbb{C}$. ☼

Recall, that the existence of a right adjoint to a given functor ensures that the functor in question preserves *all* colimits. Hence a sufficient condition for the preservation of pushouts by the graphing functor is the existence of a right adjoint to the graphing functor $\Gamma^{\mathbb{C}}\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$; as a fact, such a right adjoint exists if and only if $\mathbb{C}$ allows to classify partial maps.

☼ Definition 2.39 (Partial map classification) Let $\mathbb{C}$ be a category with pullbacks (along monomorphisms) and let $\eta_B\colon B \rightarrowtail \mathfrak{L}B$ be a monomorphism. Then the monomorphism η_B *classifies partial maps into* B if for each $\mathbb{C}$-arrow $f\colon X \to B$ and each $\mathbb{C}$-monomorphism $m\colon X \rightarrowtail Y$, there exists a unique morphism $\tilde{f}\colon Y \to \mathfrak{L}B$ making $B \leftarrow f - X \rightarrowtail m \rightarrow Y$ a pullback of $B \rightarrowtail \eta_B \rightarrow \mathfrak{L}B \leftarrow \tilde{f} - Y$, thus giving rise to the pullback square $\begin{smallmatrix} X & \to & Y \\ B & \to & \mathfrak{L}B \end{smallmatrix}$.

$$\forall A \in \mathbb{C}.\, \exists\, \eta_A\colon A \rightarrowtail \mathfrak{L}A.\, \forall A \xleftarrow{f} X \rightarrowtail m \rightarrow Y.\, \exists!\, \tilde{f}\colon Y \to \mathfrak{L}A.$$

$$\begin{array}{ccc} X & \xrightarrow{m} & Y \\ {\scriptstyle f}\downarrow & \lrcorner & \downarrow{\scriptstyle \tilde{f}} \\ A & \xrightarrow[\eta_A]{} & \mathfrak{L}A \end{array}$$

Figure 34: Partial map classifier(s)

The category $\mathbb{C}$ *has partial map classifiers* if for each object $A \in \mathbb{C}$ there exists an object $\mathfrak{L}A \in \mathbb{C}$ and a monomorphism $\eta_A\colon A \rightarrowtail \mathfrak{L}A$ such that η_A classifies partial maps into A (see Figure 34). ☼

A partial map classifier of a set $A \in \mathbb{Sets}$ is nothing else but the well-known inclusion of the set A into its "lifting" $A \cup \{\bot\}$, which allows to represent

[25] The (terminology of) this definition is linked to the concept of the *function graph* of any function $f\colon A \to B$ in $\mathbb{Sets}$, namely the relation $\Gamma_f = \{(a, f(a)) \mid a \in A\} \subseteq A \times B$, which is nothing else but the subset which corresponds to the inclusion map $\langle \mathrm{id}_A, f\rangle\colon \Gamma_f \rightarrowtail A \times B$; however the latter is essentially the same as the *pair* $\langle \mathrm{id}_A, f\rangle$ (and hence also the notation is "sound").

each partial map $f\colon X \rightharpoonup A$ by the obvious total map $\tilde{f}\colon X \to A \cup \{\bot\}$, namely $\tilde{f}(x) = f(x)$ whenever f is defined on $x \in X$ and $\tilde{f}(x) = \bot$ otherwise. As mentioned before, existence of partial map classifiers can be captured succinctly in terms of right adjoints to the graphing functor.

◎ Fact 2.40 (Partial map classification) Let $\mathbb{C}$ be a category with pullbacks (along monomorphisms), and let $\mathrm{Par}(\mathbb{C})$ the associated partial map category.

Then $\mathbb{C}$ has partial map classifiers if and only if the graphing functor $\Gamma^{\mathbb{C}}\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$ has a right adjoint.

Summarizing the above discussion of the role of partial map classifiers, a DPO rewriting step $A \models\!\!\!=\!\!\!\Rightarrow_{\langle q,m\rangle} B$ in the category $\mathbb{C}$ with a non-consuming rule $q = L \leftarrow_{\mathrm{id}} L \rightarrow_{\beta} R$ is a witness for a SPO rewriting step $A \models\!\!=\!\!\Rightarrow_{\langle [\beta],[m]\rangle} B$ if and only if the second pushout of the witnessing DPO direct derivation diagram is preserved by the graphing functor $\Gamma^{\mathbb{C}}\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$. A short sufficient condition that ensures that this is always the case is the existence of a right adjoint to the graphing functor or – equivalently – existence of partial map classifiers in the category $\mathbb{C}$. This is the main role of partial map classifiers in the context of this thesis. However, they will also feature in the next subsection, which will complete the explanation of why DPO-rewriting is a special case of SPO-rewriting in the sense of Fact 2.33 in any adhesive category.

2.5.2 The role of final pullback complements

The second crucial property of adhesive categories which ensures that each double pushout step is also a single pushout step (Fact 2.33) concerns pushout complements. However uniqueness of pushout complements by itself (Fact 2.22) is *not* sufficient. More important is that pushout complements in an adhesive category such as the category of $\mathtt{Graphs}$ or some topos, form pullback complements in the sense of [Dyckhoff and Tholen, 1987], which here will be called *final* pullback complements. In less precise terms, final pullback complements are universal solutions for the problem of finding a pullback complement for a given pair of composable morphisms. As a rough approximation, they can be thought of as a generalization of (relative) complements in a subset poset.

Recall that an (arbitrary) pullback complement of a pair of composable morphisms $C \leftarrow_h B \leftarrow_a A$ is another pair $C \leftarrow_d D \leftarrow_k A$ of composable morphisms that gives rise to the pullback square $\begin{smallmatrix} B & \downarrow & \leftarrow \\ C & \leftarrow & \downarrow \end{smallmatrix}\!\!\begin{smallmatrix} \lrcorner \\ \downarrow \end{smallmatrix}\!\!\begin{smallmatrix} A \\ D \end{smallmatrix}$. However, as might have become clear by now, the case in which $a\colon A \rightarrowtail B$ is monic is of special interest in the context of this thesis. Further, given such a pair $C \leftarrow_h B \leftarrow_a A$ with monic a, one might want to consider only those pullback complements

$C \leftarrow d \prec D \leftarrow k - A$ with monic d, and call them *monic* pullback complements. Finally, using the language of [Kennaway, 1990], one might ask for a "largest" such pullback complement. This leads to the following variation of the pullback complements of [Dyckhoff and Tholen, 1987].

☼ Definition 2.41 (Mono-final pullback complements) Let $\mathbb{C}$ be a category, let $C \leftarrow h - B \leftarrow a \prec A$ be pair of composable morphisms with monic a.

A *mono-final pullback complement* (MFPBC) *of* $C \leftarrow h - B \leftarrow a \prec A$ is a pullback complement $C \leftarrow y \prec Y \leftarrow x - A$ with monic y such that for any monomorphism $e\colon E \rightarrowtail C$, pullback $B \leftarrow p \prec P \rightarrow q - E$ of $B \rightarrow h - C \leftarrow e \prec E$, and morphism $r\colon (P \succ p \rightarrow \cdot) \rightarrowtail (A \succ a \rightarrow \cdot)$ in $\mathbb{C}\downarrow B$, there is a morphism $z\colon (E \succ e \rightarrow \cdot) \rightarrowtail (Y \succ y \rightarrow \cdot)$ in $\mathbb{C}\downarrow C$ satisfying $h \circ r = z \circ x$, thus giving rise to a pullback square $\begin{smallmatrix} A & \leftarrow \\ Y & \downarrow \end{smallmatrix} \begin{smallmatrix} \lrcorner & \\ \leftarrow & \downarrow \end{smallmatrix} \begin{smallmatrix} P \\ E \end{smallmatrix}$ in $\mathbb{C}$ (see also Figure 35). ☼

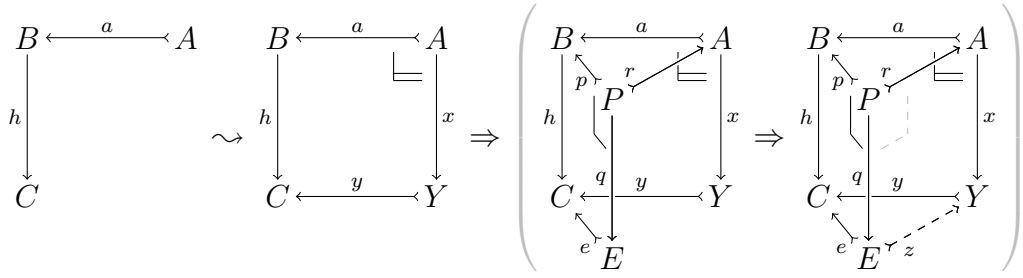


Figure 35: Mono-final pullback complement

Mono-final pullback complements are unique up to isomorphism, i.e. whenever $C \leftarrow y \prec Y \leftarrow x - A$ and $C \leftarrow y' \prec Y' \leftarrow x' - A$ are MFPBCs of $C \leftarrow h - B \leftarrow a \prec A$, then there exists an isomorphism $i\colon Y \rightarrow\!\!\!\!\!\sim Y'$ such that $y = y' \circ i$ and $x' = i \circ x$. Indeed, uniqueness of pushout complements in adhesive categories (Fact 2.22) is proved in [Lack and Sobociński, 2005] using the fact that pushout complements for a pair of composable morphisms $C \leftarrow h - B \leftarrow a \prec A$ with monic a are always mono-final pullback complements.

In a wide range of categories, including $\mathbb{G}\mathtt{raphs}$, and all topoi and Heyting categories, mono-final pullback complements are closely related to right adjoints to inverse image functors. More detailed, for a composable pair $C \leftarrow h - B \leftarrow a \prec A$ with MFPBC $C \leftarrow y \prec Y \leftarrow x - A$, the subobject $[y] \in \mathrm{Sub}(C)$ is the image of $[a] \in \mathrm{Sub}(B)$ under the the right adjoint $\forall_h\colon \mathrm{Sub}(B) \rightarrow \mathrm{Sub}(C)$ of the pre-image functor $h^{-1}\colon \mathrm{Sub}(C) \rightarrow \mathrm{Sub}(B)$; in other words $y \cong \forall_h(a)$ in $\mathrm{Sub}(C)$.

Conversely, given the pair $C \leftarrow h - B \leftarrow a \prec A$, if the counit of the adjunction at a is an isomorphism, i.e. if $\mathbb{c}_a\colon h^{-1} \circ \forall_h(a) \rightarrow\!\!\!\!\!\sim a$ is invertible where $\mathbb{c}$ is

the counit of the adjunction $h^{-1} \dashv \forall_h$, then there exists a mono-final pullback complement $C \leftarrow y \prec Y \leftarrow x - A$ of $C \leftarrow h - B \leftarrow a \prec A$ with $y \cong \forall_h(a)$. Summarizing, this yields the following facts (cf. [Dyckhoff and Tholen, 1987, Theorem 4.4]).

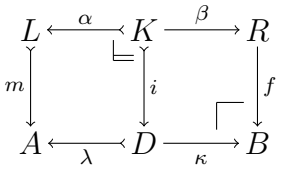
◎ Fact 2.42 (Existence of mono-final pullback complements) Let $\mathbb{C}$ be a category with pullbacks along monomorphisms, let $f\colon X \to Y$ in $\mathbb{C}$ be a morphism, and let $\forall_f\colon \mathrm{Sub}(X) \to \mathrm{Sub}(Y)$ be the right adjoint to the inverse image functor $f^{-1}\colon \mathrm{Sub}(Y) \to \mathrm{Sub}(X)$ with counit $\mathfrak{c}\colon f^{-1} \circ \forall_f \to \mathrm{id}_{\mathrm{Sub}(X)}$. Then the following are true.

1. For any monomorphism $a\colon A \rightarrowtail X$ over X, there exists a mono-final pullback complement for the composable pair $Y \leftarrow f - X \leftarrow a \prec A$ if and only if the morphism $\mathfrak{c}_a\colon f^{-1} \circ \forall_f(a) \rightarrowtail a$ is an isomorphism.

2. If $f\colon X \rightarrowtail Y$ is monic, then $\mathfrak{c}$ is an isomorphism, i.e. $\mathfrak{c}_a$ is an isomorphism for every monomorphism $a\colon A \rightarrowtail X$ over X.

Hence, if the right adjoint $\forall_f$ to pre-image functors f^{-1} exists for all morphisms f – as it happens to be the case in the category of $\mathbb{G}\mathrm{raphs}$ and in any topos or Heyting category – there exists an MFPBC for every pair of composable monomorphisms. As a consequence, in a suitable category $\mathbb{C}$, for the case of monic matches,[26] single pushout rewriting in the partial map category $\mathrm{Par}(\mathbb{C})$ has the following equivalent description in $\mathbb{C}$ – a fact which can be proved along the lines of [Kennaway, 1990].

◎ Fact 2.43 (Single pushout rewriting with monic matches)



Let $\mathbb{C}$ be a category with pullbacks along monomorphisms. Further let $[\alpha_{(K)}\beta\rangle\colon L \rightharpoonup R$ in $\mathrm{Par}(\mathbb{C})$ be a partial map and let $[m\rangle\colon L \rightarrowtail A$ be a (total) monomorphism. Moreover suppose that there exists a diagram in $\mathbb{C}$ as displayed, i.e. let $A \leftarrow \lambda \prec D \leftarrow i \prec K$ be the MFPBC of $A \leftarrow m \prec L \leftarrow \alpha \prec K$, and let $D - \kappa \rightarrow B \leftarrow f - R$ be the pushout of $D \leftarrow i \prec K \succ \beta \rightarrow R$.

Then the displayed diagram witnesses the single pushout rewriting step $A \models\!\!\!\Rightarrow^{\langle[\alpha,\beta\rangle,[m\rangle\rangle} B$ if and only if the right square is a pushout in the partial map category, which means that the cospan $A - [\lambda,\kappa\rangle \rightharpoonup B \leftharpoonup [f\rangle - R$ is the pushout of the span $A \leftharpoonup [m\rangle - L - [\alpha,\beta\rangle \rightharpoonup R$ in $\mathrm{Par}(\mathbb{C})$ if and only if $D - [\kappa\rangle \rightharpoonup B \leftharpoonup [f\rangle - R$ is a pushout of $D \leftharpoonup [i\rangle - K - [\beta\rangle \rightharpoonup R$ in $\mathrm{Par}(\mathbb{C})$. In particular, the left square corresponds to a pushout in the partial map category, i.e. $A - [\lambda,\mathrm{id}\rangle \rightharpoonup D \leftharpoonup [i\rangle - K$ is a pushout of $A \leftharpoonup [m\rangle - L - [\alpha,\mathrm{id}\rangle \rightharpoonup K$ in the partial map category $\mathrm{Par}(\mathbb{C})$.

[26] Requiring monic matches amounts to taking the lluf category of (total) monomorphisms for $\mathbb{M}$ in Definition 2.2.

The existence of MFPBCs for pairs of composable monomorphisms is ensured if the ambient category has partial map classifiers – a fact which follows from the proof for Corollary 4.6 of [Dyckhoff and Tholen, 1987].

Fact 2.44 (Final pullback complements from partial map classifiers) Let $\mathbb{C}$ be a category with pullbacks (along monomorphisms). If $\mathbb{C}$ has partial map classifiers then it also has mono-final pullback complements for pairs of composable monomorphisms.

Now all relevant concepts are introduced to address a general version of Fact 2.33 which sums up the central relation between single and double pushout rewriting that will be relevant for this thesis.

2.5.3 Double pushout is reversible single pushout

Given a linear rule $q = L \leftarrow\!\alpha\!\prec K \succ\!\beta\!\rightarrow R$ in some adhesive category, then every double pushout derivation $A \models\!\!\models\!\langle q,m\rangle\!\Rightarrow B$ with a monic match m is actually witnessed by a diagram as in Fact 2.43. The reason is that the first square of the double diagram has been obtained by a pushout complement for $A \leftarrow\!m\!- L \leftarrow\!\alpha\!\prec K$, and every pushout complement for a $A \leftarrow\!m\!- L \leftarrow\!\alpha\!\prec K$ is also a mono-final pullback complement (see [Lack and Sobociński, 2005, Lemma 2.8]); the second square is a pushout by assumption.

In the other direction, when the rule q is interpreted as a partial map $\varrho_q = [\alpha,\beta\rangle\colon L \rightharpoonup R$ in $\mathrm{Par}(\mathbb{C})$, every single pushout derivation $A \models\!\langle \varrho_q,[n]\rangle\!\Rightarrow B$ with monic n corresponds to a double pushout rewriting step $A \models\!\!\models\!\langle q,n\rangle\!\Rightarrow B$ if and only if it *reversible*, i.e. if there is an opposite derivation $B \models\!\langle \varrho_q^\circ,[f]\rangle\!\Rightarrow A$ back to A where f is the back-match and $\varrho_q^\circ = [\beta,\alpha\rangle\colon R \rightharpoonup L$ is the partial function of the opposite rule q° as shown in Figure 36.

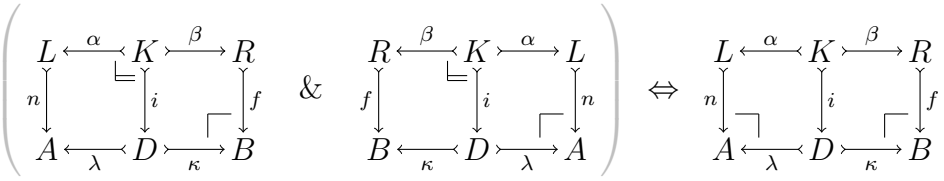


Figure 36: The double pushout counterpart of a reversible single pushout step

This fundamental relationship between double pushout rewriting in a category $\mathbb{C}$ and single pushout rewriting in the associated partial map category $\mathrm{Par}(\mathbb{C})$ is discussed for the case of $\mathbb{G}\mathrm{raphs}$ in [Ehrig et al., 1997]. The definition of *weakly* adhesive categories that will be given in Section 3 ensures that

single and double pushout remain closely related in the above manner. Some background on alternative proposals for generalizations of adhesive categories is given next.

2.6 Variants of adhesivity

The remainder of this section gives an overview of proposals to generalize the concept of adhesivity. The proposed variations of adhesivity are motivated by the fact that there are examples of categories that occur passim in computer science and mathematics and are generally accepted to be "well-behaved" or "rich" but nevertheless are not adhesive. The two main directions for solving this mismatch are the restriction to a suitable class of monomorphisms (instead of *all* monomorphisms) and the weakening of the conditions that pushouts (along monomorphisms of such a class) are required to satisfy.

The first line of thought is already present in [Lack and Sobociński, 2005], where the class of *all* monomorphisms in the definition of adhesivity is replaced by the class of all *regular* monomorphisms. The resulting class of categories were dubbed *quasi-adhesive*. However the subsequent paper [Johnstone et al., 2007] demonstrated that this generalization does not reach far enough to include the category of simple graphs (see also Example 2.47).

However, by combining the idea of replacing the class of *all* monomorphisms by a suitable class $\mathscr{M}$ of monomorphisms with the weakening of the conditions that pushouts along monomorphisms in $\mathscr{M}$ must satisfy it becomes possible to obtain a weakened notion of adhesivity that embraces not only the category of simple graphs but also the category of topological spaces, all quasi-topoi and attributed[27] graphs.

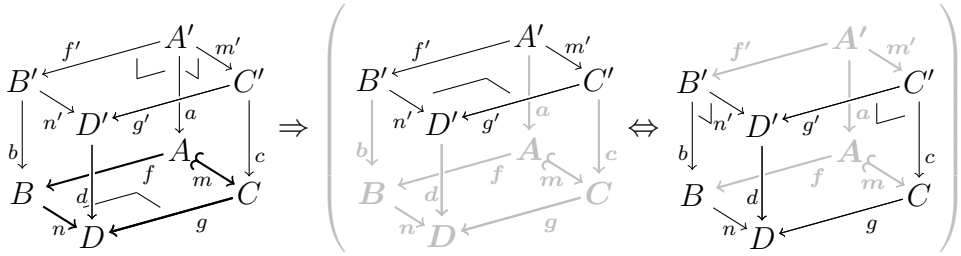


Figure 37: $\mathscr{M}$-weak VK square property (either $f \in \mathscr{M}$ or $\{b, c, d\} \subseteq \mathscr{M}$)

The notion of weak adhesive HLR category w.r.t. $\mathscr{M}$ follows the latter

[27] See [Ehrig et al., 2004c] for a presentation of attributed, typed graphs.

strategy; the main idea of these categories is to replace the requirement that pushouts along monomorphisms in $\mathscr{M}$ are Van Kampen squares (see Figure 25) by a weak Van Kampen square property as illustrated in Figure 37. The full definition of [Ehrig and Prange, 2006] is as follows (up to some minor changes in presentation).

☼ Definition 2.45 (Weak adhesive HLR w.r.t. $\mathscr{M}$) A category $\mathbb{C}$ with a morphism class $\mathscr{M}$ is a *weak adhesive* HLR *category*, if

1. $\mathscr{M}$ is a class of monomorphisms closed under isomorphisms, composition ($f\colon A \to B \in \mathscr{M}, g\colon B \to C \in M \Rightarrow g \circ f \in \mathscr{M}$) and decomposition ($g \circ f \in \mathscr{M}, g \in \mathscr{M} \Rightarrow f \in \mathscr{M}$),
2. $\mathbb{C}$ has pushouts and pullbacks along $\mathscr{M}$-morphisms and $\mathscr{M}$-morphisms are closed under pushouts and pullbacks,
3. pushouts in $\mathbb{C}$ along $\mathscr{M}$-morphisms are $\mathscr{M}$-weak VK squares,

i.e. the Van Kampen square property holds for all commutative cubes with $m \in \mathscr{M}$ and moreover $f \in \mathscr{M}$ or $b, c, d \in \mathscr{M}$ (see also Figure 37). ☼

Examples of weak adhesive HLR include all (quasi-)adhesive categories and also the categories of simple graphs and topological spaces (see Example 3.2). One of the properties of (quasi-)adhesive categories that does not have a counterpart in weak adhesive HLR categories is the "relatively rare property"[28] of having effective unions.

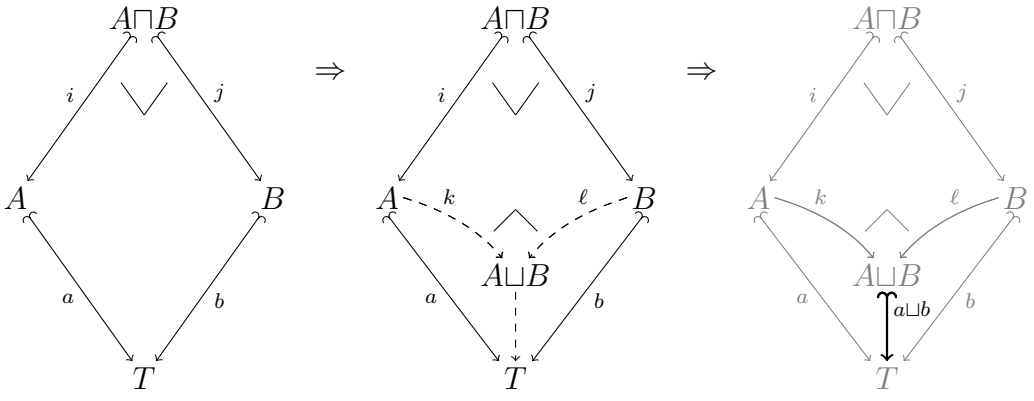


Figure 38: Effective unions

[28] This quote is taken from [Barr, 1987].

☼ Definition 2.46 ($\mathscr{M}$-effective unions) Let $\mathbb{C}$ be a category that comes equipped with a class $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$ of monomorphisms in $\mathbb{C}$. Then the category $\mathbb{C}$ *has effective unions* if for any $\mathbb{C}$-cospan $A \hookleftarrow a \rightarrow T \leftarrow b \hookrightarrow B$ where a and b are monomorphisms in $\mathscr{M}$, and for any pullback $A \leftarrow i \hookrightarrow A \sqcap B \hookleftarrow j \rightarrow B$ of $A \hookleftarrow a \rightarrow T \leftarrow b \hookrightarrow B$ with $i, j \in \mathscr{M}$, a pushout $A \overset{k}{\rightarrow} A \sqcup B \overset{\ell}{\leftarrow} B$ of $A \leftarrow i \hookrightarrow A \sqcap B \hookleftarrow j \rightarrow B$ exists and the mediating morphism $z \colon A \sqcup B \to T$ satisfying $a = k \fatsemi z$ and $z \circ \ell = b$ is a monomorphism in $\mathscr{M}$; in such a situation z is often written as $a \sqcup b$ (see Figure 38). ☼

❦ Example 2.47 (Counterexample for effective unions) Consider the category of simple graphs $\mathtt{sGraphs}$. Its objects are pairs $\langle V, E \rangle$ where V is a set of *vertices* and $E \subseteq V \times V$ is a binary relation and the elements this relation E are called *edges*. Further a morphism $f \colon \langle V, E \rangle \to \langle W, F \rangle$ between two graphs $\langle V, E \rangle$ and $\langle W, F \rangle$ is a function $f \colon V \to W$ which preserves edges, i.e. $\langle u, v \rangle \in E$ implies $\langle f(u), f(v) \rangle \in F$ for all $u, v \in V$. As two basic facts, such a function is a monomorphism if and only if it is injective and it is a *regular* monomorphism if and only if it is injective and reflects edges, i.e. $\langle f(u), f(v) \rangle \in F$ implies $\langle u, v \rangle \in E$ for all $u, v \in V$.

This category of simple graphs does not have regular-effective unions. Figure 39 illustrates a counterexample, which is based on [Johnstone et al., 2007]. The inclusions of the nodes ⓤ and ⓥ into the graph ⓤ→ⓥ are regular

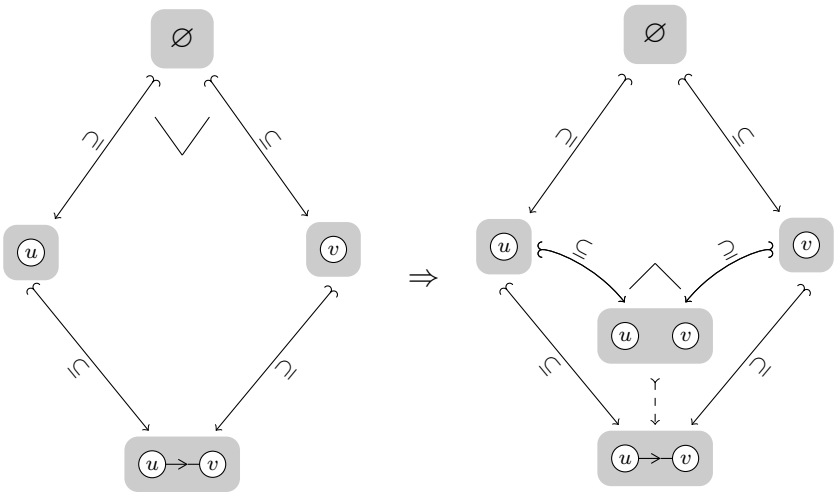


Figure 39: Counterexample to regular effective unions

monomorphisms; their intersection is the empty graph (see the left hand diagram in Figure 39). Taking the pushout over this empty graph yields the

coproduct, namely the two node graph ⓤ ⓥ. Although the latter graph is a subgraph of ⓤ→ⓥ, the inclusion morphism does not reflect the edge, and hence is not regular. ☙

Though each weak adhesive HLR category w.r.t. $\mathscr{M}$ has an associated category of partial maps where the domain of definition of each partial map is some $\mathscr{M}$-morphism, it is not clear yet whether pushouts along $\mathscr{M}$-morphisms are preserved by the embedding into this partial map category. If this is not the case, then not every double pushout step is guaranteed to be also a single pushout step in the category of $\mathscr{M}$-partial maps and the tight connection between single and double pushout rewriting know from the category of graphs would be lost. This issue will be discussed in more detail in Section 8. The category of $\mathscr{M}$-partial maps is well-defined and is characterized as the largest lluf subcategory of $\mathrm{Par}(\mathbb{C})$ which contains all partial maps with an $\mathscr{M}$-morphism as "domain of definition". This is spelled out in the following definition.

☼ Definition 2.48 (Category of $\mathscr{M}$-partial maps) Let $\mathbb{C}$ be a category. A *dominion in* $\mathbb{C}$ [Rosolini, 1986] is a class of monomorphisms $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$ such that

- ⋇ pullbacks along $\mathscr{M}$-morphisms exist, and the class $\mathscr{M}$ is stable under pullback;
- ⋇ the class $\mathscr{M}$ is closed under isomorphisms and composition (with $\mathscr{M}$-morphisms).

Such a class $\mathscr{M}$ is also called an *admissible class* of monomorphisms or subobjects [Robinson and Rosolini, 1988].

Given a dominion $\mathscr{M}$ in $\mathbb{C}$, the category of *$\mathscr{M}$-partial maps*, written $\mathrm{Par}(\mathbb{C},\mathscr{M})$, has the same objects as $\mathbb{C}$ and for a given pair of objects $A, B \in \mathbb{C}$ the morphisms from A to B are the homset

$$\mathrm{Par}(\mathbb{C},\mathscr{M})(A,B) = \left\{ [m,f\rangle \colon A \rightharpoonup B \;\middle|\; \begin{array}{l} A \leftarrow m \triangleright\, A' \,{-}f\rightarrow B \text{ is} \\ \text{a } \mathbb{C}\text{-span with } m \in \mathscr{M} \end{array} \right\}$$

where $[m,f\rangle$ is the isomorphism class of $A \leftarrow m \triangleright\, A' \,{-}f\rightarrow B$ as in Definition 2.32. Identities and compositions are inherited from the partial map category $\mathrm{Par}(\mathbb{C})$. Finally also a *graphing functor* $\Gamma_{\mathscr{M}} \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C},\mathscr{M})$ can be defined as in Definition 2.38. ☼

This definition will be central for the theoretical foundations of this thesis.

Concurrent Semantics of Rewriting

Weak adhesivity and concurrency in rewriting

3

The underlying idea of a computation up to concurrency in single or double pushout rewriting can be related to the theory of partially commutative monoids – a.k.a. *trace monoids* or Mazurkiewicz traces – as follows. The starting point of trace monoids is an alphabet $\Sigma = \{a, b, c, \ldots\}$ of letters, which in the present context might be thought of as representing (atomic) actions in some system (the structure of which is abstracted away from). Additionally, this set Σ comes equipped with an irreflexive, symmetric independence relation $I \subseteq \Sigma \times \Sigma$, thus yielding an *independence alphabet* $\langle \Sigma, I \rangle$; two letters $a, b \in \Sigma$ are said to be *independent* of each other if $\langle a, b \rangle \in I$ holds true. The fact that a and b are independent of each other might be meant to express that in a given system, the two actions which correspond to the elements a and b are independent of each other, i.e. there do not exist any *causal* dependencies between them. In such a situation, the order in which these actions may take place in some run of the system is completely accidental. Hence, on the formal side, the words ab and ba are considered equivalent, and more general, two words over the alphabet Σ are equivalent if one can be obtained from the other by (repeatedly) switching neighboring pairs of independent letters, and thus each string describes a whole equivalence class of strings, namely a *Mazurkiewicz trace*.

This idea carries over to pushout based rewriting, by replacing '*letter*' with '*direct derivation diagram*' and '*independent*' with '*sequential-independent*', thus yielding a theory of concurrency in pushout based rewriting. This analogy is in accordance with the use of derivation diagrams as a model for atomic, local changes in event based systems: two events that correspond to a pair of sequential-independent direct derivation diagrams should not be causally dependent on each other (if the system model is supposed to be sound). The weak point of the analogy is the fact that – in general – pairs of sequential-independent direct derivation diagrams can be switched in more than one way: this is where new insights can be gained from pushout based rewriting. One advantage of the more general approach is that it allows to represent the structure of system states and it moreover incorporates the principle of local change; the latter principle also accounts for the observation that cause and effect are usually close to each other in terms of space (and time).

Before the actual investigation of concurrency in systems that are modeled using single or double pushout rewriting starts, a category with convenient properties is fixed in advance to ensure that the resulting concurrent semantics

of systems does not become unduly complicated. Using the category of $\mathbb{G}\mathrm{raphs}$ or any adhesive category or topos yields appropriate results; however the goal is to work in a framework that is general enough to include also the category $\mathbb{sG}\mathrm{raphs}$ – having the widely used simple graphs[29] as objects – and the category of topological spaces $\mathbb{T}\mathrm{op}$, which pervades mathematics and is also relevant to theoretical computer science where it occurs in the guise of the category $\mathbb{P}\mathrm{re}$ of preorders and order preserving maps, which is a full subcategory of $\mathbb{T}\mathrm{op}$. All these example categories will be accounted for in the framework of weakly adhesive categories.

Overview of the section. The main objective of this section is to introduce so-called *switch-equivalence classes* of sequences of double pushout rewriting steps (Definition 3.9) in analogy to the theory Mazurkiewicz traces, however on a more general, category theoretical level. A switch-equivalence class is a "generalized trace" and formalizes a computation up to concurrency. This concept is the starting point for further investigations in Section 4.

The abstract framework is laid out in Section 3.1, which defines weakly $\mathscr{M}$-adhesive categories and discusses the paradigmatic examples, namely two different categories of graphs and the category of topological spaces. Some remarks on the relation with the (quasi-)adhesive categories of [Lack and Sobociński, 2005] are given as well.

The main definition of double pushout traces is given in Section 3.2. Building on the analogy with Mazurkiewicz traces, double pushout derivations are presented as the counterpart to strings; the natural independence relation is given by sequential independence. The main technical complication is the description of the *canonical* way to switch a pair of sequential-independent direct derivations. With this notion at hand, double pushout traces arise as the obvious equivalence classes of double pushout rewriting sequences which arise from each other by (repeated) switching of pairs of sequential-independent direct derivations.

3.1 Weak adhesivity

Retaining the idea of the existence of "enough" pushouts with "good behaviour" from adhesive categories [Lack and Sobociński, 2005], two stability criteria are

[29] The category $\mathbb{sG}\mathrm{raphs}$ has binary endorelations $E \subseteq V \times V$ as objects, and a morphism from $E \subseteq V \times V$ to $F \subseteq W \times W$ is a function $\varphi\colon V \to W$ which preserves the relation, i.e. $\langle u, v\rangle \in E$ implies $\langle \varphi(u), \varphi(v)\rangle \in F$ for all $u, v \in V$.

imposed on pushouts, namely *universality* and *preservation by the graphing functor* as described below in Definition 3.1. The resulting concept is dubbed *weak adhesivity* since every adhesive category will turn out to be weakly adhesive and the similarities between weakly adhesive and weak adhesive HLR categories outweigh their differences as argued in Section 8.

As the properties of pushouts that make a category $\mathbb{C}$ weakly adhesive are described in terms of the associated partial map category $\mathrm{Par}(\mathbb{C})$, it must be ensured in the first place that the latter is well-defined. Following [Rosolini, 1986, page 27], the definition of categories of partial maps uses a category $\mathbb{C}$ with a *dominion* of monomorphisms where a dominion is a class $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$ of monomorphisms in $\mathbb{C}$ which contains all identities, is closed under composition, and additionally pullbacks along morphisms in $\mathscr{M}$ exist and $\mathscr{M}$ is stable under pullbacks; the latter means that for each co-span $A \xrightarrow{f} D \xleftarrow{m} M$ with $m \in \mathscr{M}$, a pullback $A \xleftarrow{n} N \xrightarrow{g} M$ exists, yielding a *pullback square* $\begin{smallmatrix} N & \xrightarrow{} & M \\ A & \xrightarrow{} & D \end{smallmatrix}$, and n is again a morphism in $\mathscr{M}$. Morphisms that belong to the class $\mathscr{M}$ are called $\mathscr{M}$-morphisms. Dominions are also known as admissible classes of monomorphisms (cf. [Robinson and Rosolini, 1988]).

The trivial dominion $\mathscr{Iso}$ is the class of all isomorphisms, the largest possible dominion $\mathscr{Mono}$ contains all monomorphisms; other typical examples include the collection of regular monomorphisms[30] $\mathscr{Reg}$ in $\mathtt{sGraphs}$ or any quasi-topos [Penon, 1977], and the classes $\mathscr{Open}$ and $\mathscr{Closed}$ in the category of topological spaces, which contain the open and closed embeddings, respectively (cf. [Niefield, 1982]). In the context of this thesis, dominions are used as a replacement for the class of all monomorphism if the class of all monomorphisms does not have all desired properties, e.g. if pushouts along arbitrary monomorphisms are not pushouts in the partial map category as illustrated in Example 2.37.

Speaking in terms of the metaphor of the introduction, such a dominion $\mathscr{M}$ in a category $\mathbb{C}$ provides embeddings and a "clipping"-operation, which was used to circumscribe pullbacks along $\mathscr{M}$-morphisms. Moreover, "gluing" of a span $B \xleftarrow{f} A \xrightarrow{g} C$ was used to paraphrase pushouts, and indeed, weakly $\mathscr{M}$-adhesive categories will have pushouts of such spans by definition, at least if one of f or g are $\mathscr{M}$-morphisms – a condition which was added to obtain a clearer connection to adhesive categories [Lack and Sobociński, 2005] and

[30] A monomorphisms $m\colon A \rightarrowtail B$ is regular if there exist two arrows $f, g\colon B \rightrightarrows C$ such that m is the equalizer of f and g. In $\mathtt{sGraphs}$, a monomorphism is regular if it *reflects* edges, i.e. a morphism $m\colon V \to W$ between simple graphs $V \subseteq E \times E$ and $F \subseteq W \times W$ belongs to $\mathscr{Reg}$ if and only if m is injective and $\langle m(u), m(v)\rangle \in F$ implies $\langle u, v\rangle \in E$ for all $u, v \in V$.

weak adhesive HLR categories [Ehrig and Prange, 2006]. However, the typical example categories such as (simple) graphs actually have all pushouts (see also Section 7.2).

The main requirement on pushouts in weakly $\mathscr{M}$-adhesive categories will be "well-behaved" w.r.t. pullbacks: first, pushouts (along $\mathscr{M}$-morphisms) must also be pushouts in the category of $\mathscr{M}$-partial maps (in the introduction, this was described metaphorically as the compatibility of "gluing" and "clipping"); and second, pushouts of pairs of $\mathscr{M}$-morphisms $B \leftarrow m \rightarrowtail A \leftarrow n \rightarrow C$ must be universal – a property which can be thought of as a generalization of the fact that the preimage-operation preserves unions[31] in the category of $\mathbb{Sets}$.

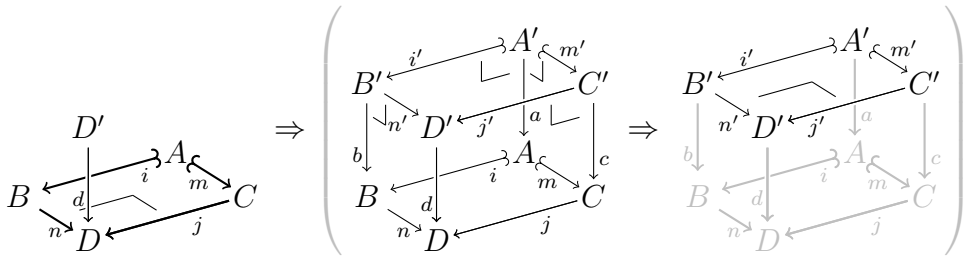
☼ Definition 3.1 (Weakly $\mathscr{M}$-adhesive category) Let $\mathbb{C}$ be category with a dominion $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$.

Then the category $\mathbb{C}$ is *weakly $\mathscr{M}$-adhesive* if

1. pushouts along $\mathscr{M}$-morphisms exist, which means that for each span $B \leftarrow f - A \leftarrowtail m \rightarrow C$ with $m \in \mathscr{M}$, a pushout $B - n \rightarrow D \leftarrow g - C$ exists, yielding a *pushout square* $\begin{smallmatrix} B & \leftarrow & A \\ \downarrow & & \downarrow \\ D & \leftarrow & C \end{smallmatrix}$;

2. pushouts along $\mathscr{M}$-morphisms remain pushouts after embedding them into the category of $\mathscr{M}$-partial maps, i.e. pushouts are preserved by the graphing functor $\Gamma_{\mathscr{M}} \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C}, \mathscr{M})$;[32]

3. pushouts of pairs of $\mathscr{M}$-morphisms are universal (or *stable under pullback*), i.e. in each commutative cube over a pushout square $\begin{smallmatrix} B & \leftarrow & A \\ & \searrow & \\ & D & \leftarrow & C \end{smallmatrix}$ having pullback squares as lateral faces as shown in the middle diagram in the display below, the top face is a pushout square.



[31] In $\mathbb{Sets}$, the equation $d^{-1}(B \cup C) = d^{-1}(B) \cup d^{-1}(C)$ holds for all functions $d \colon D' \to D$ and pairs of subsets $B, C \subseteq D'$.

[32] That $\Gamma \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C}, \mathscr{M})$ preserves some pushout $B - n \rightarrow D \leftarrow j - C$ of a span $B \leftarrow i - A - m \rightarrow C$ means that $B - \Gamma n \rightharpoonup D \leftharpoonup \Gamma j - C$ is a pushout of the pair $B \leftharpoonup \Gamma i - A - \Gamma m \rightharpoonup C$ in the partial map category $\mathrm{Par}(\mathbb{C}, \mathscr{M})$.

A category is called *weakly adhesive* if it it is weakly $\mathscr{M}ono$-adhesive where $\mathscr{M}ono$ is the class of *all* monomorphisms, and it is called *weakly quasi-adhesive* if it is weakly $\mathscr{R}eg$-adhesive where $\mathscr{R}eg$ is the class of all regular monomorphisms.

Though a detailed discussion of the relations of this version of adhesivity to the other variants is deferred to Section 8, it might be worth to mention at this point that all (quasi-)adhesive categories are weakly (quasi-)adhesive, and all weakly $\mathscr{M}$-adhesive categories are weak adhesive HLR categories w.r.t. $\mathscr{M}$.

Before turning to the investigation of the concept of concurrent computation in such a category, consider the following examples of weakly $\mathscr{M}$-adhesive categories, which are meant to illustrate the role of the "parameter" $\mathscr{M}$.

Example 3.2 (Simple graphs and finite topological spaces) The category of simple graphs, is not weakly adhesive because of Example 2.37. However it is weakly $\mathscr{R}eg$-adhesive where $\mathscr{R}eg$ coincides with the class of edge-reflecting monomorphisms. More general, every quasi-topos is weakly $\mathscr{R}eg$-adhesive as shown in Section 9.

Next, consider the following basic facts from topology. As described in e.g. in [May, 2003], the category of finite topological spaces $\mathfrak{finTop}$ is actually isomorphic to the category of $\mathfrak{finPre}$ finite preorders and order preserving maps, which could alternatively be described as the full subcategory of $\mathfrak{sGraphs}$ which has only those graphs as objects which describe a (finite) reflexive and transitive relations.

The "specialization"-functor from $\mathfrak{finTop}$ to $\mathfrak{finPre}$ maps each topological space $\langle X, \mathcal{O}\rangle$ to the preorder $\langle X, \leq_{\mathcal{O}}\rangle$ where $\leq_{\mathcal{O}}$ is the *specialization preorder*, i.e. $x \leq_{\mathcal{O}} y$ holds for any two elements $x, y \in X$ if and only if y is contained in all open sets that contain x. Conversely, given a preorder $\langle X, \leq\rangle$, the associated topological space is $\langle X, \mathcal{O}_{\leq}\rangle$ where $\mathcal{O}_{\leq}$ is the Alexandrov topology, which means that for all $U \subseteq X$, the set U is an open in $\mathcal{O}_{\leq}$ if and only if U is upward-closed, i.e. if and only if $u \leq x$ implies $x \in U$ for all $u \in U$ and all $x \in X$.

Now, with this fact at hand, Example 2.37 can be reused to show that $\mathfrak{finTop}$ is not weakly $\mathscr{M}ono$-adhesive. Figure 40. illustrates two preorders over the set $\{x, y\}$ where reflexivity is implicit, i.e.

$\textcircled{x} \leq \textcircled{y}$ represents $P_{xy} = \Big\langle \{x, y\}, \big\{\langle x, x\rangle, \langle y, y\rangle, \langle x, y\rangle\big\}\Big\rangle$, and

$\textcircled{x} \quad \textcircled{y}$ stands for $P_0 = \Big\langle \{x, y\}, \big\{\langle x, x\rangle, \langle y, y\rangle\big\}\Big\rangle$.

As illustrated in Figure 40, the pushout of $P_{xy} \leftarrow i- P_0 -i\rightarrow P_{xy}$ in $\mathbb{C}$ yields P_{xy} again where $i\colon P_0 \to P_{xy}$ is the inclusion. To see that this square is not a

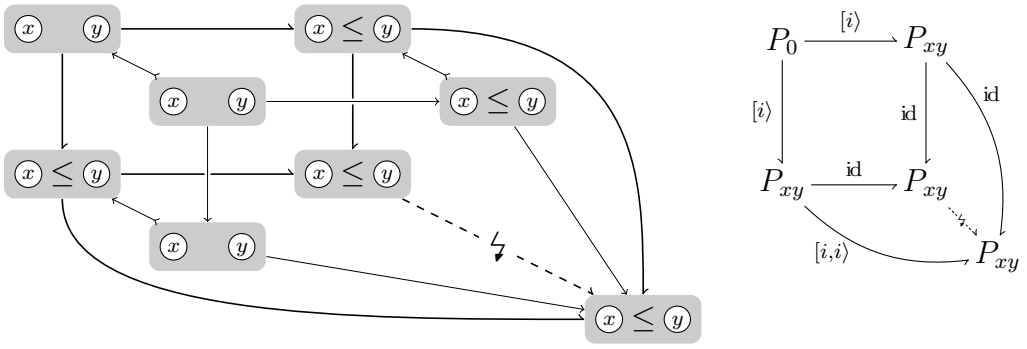


Figure 40: A pushout in $\mathbb{P}\mathtt{re}$ that is not preserved by the graphing functor

pushout in $\mathrm{Par}(\mathbb{fin}\mathbb{T}\mathtt{op})$, consider the two partial maps $\mathrm{id}_{P_{xy}} \colon P_{xy} \rightharpoonup P_{xy}$ and $[i,i\rangle \colon P_{xy} \rightharpoonup P_{xy}$ which make the boundary of the diagram of commute, i.e. the equation $[i,i\rangle \circ [i\rangle = [i\rangle = \mathrm{id}_{P_{xy}} \circ [i\rangle$ holds. However there is no mediating partial map $f \colon P_{xy} \rightharpoonup P_{xy}$ as it would have to satisfy the equation $[i,i\rangle = f = \mathrm{id}_{P_{xy}}$ which would contradict $\mathrm{id}_{P_{xy}} \neq [i,i\rangle$.

However $\mathbb{T}\mathtt{op}$ is both weakly *Open*- and *Closed*-adhesive where *Open* and *Closed* are the classes of open and closed monomorphisms, respectively, i.e. they are essentially inclusions of open and closed subspaces, respectively (see also Lemma 9.10).

Now all preparations are made to start the investigation of concurrency in sequences of direct derivation diagrams, ore more precisely of (finite) paths in the raw transition system of a given set of rules.

3.2 Traces as derivations up to concurrency

Following [Baldan, 2000], consider the case of *linear* rules, i.e. those rules that consist of two monomorphisms (in $\mathscr{M}$).

Definition 3.3 (Linearity of rules) Let $\mathbb{C}$ be a category and $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$ a class of monomorphisms; further let $q = L \leftarrow\alpha- K -\beta\rightarrow R$ be a rule.

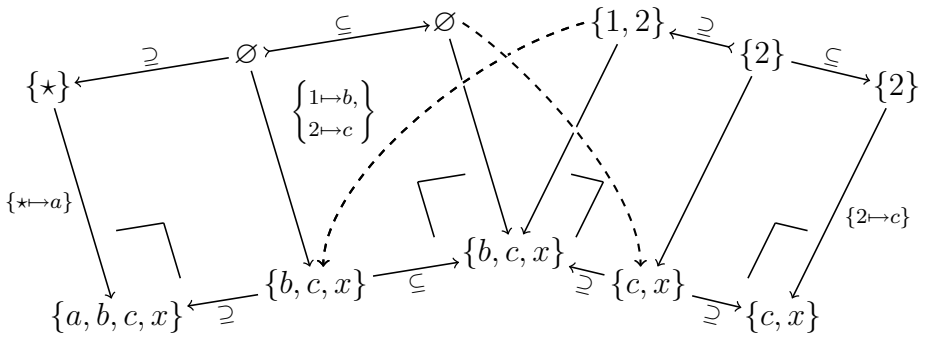
Then q is *left $\mathscr{M}$-linear* if α is an $\mathscr{M}$-morphism, and it is *$\mathscr{M}$-linear* if both α and β are $\mathscr{M}$-morphisms (if the class of monomorphisms $\mathscr{M}$ is clear from the context, the prefix '$\mathscr{M}$-' is omitted).

In the context of traces, the advantage of linear rules consist in the fact that (in a weakly $\mathscr{M}$-adhesive category[33]) any pair of sequential-independent direct

[33] In this section, also a weak adhesive HLR category w.r.t. $\mathscr{M}$ could be used instead of a

derivations has an essentially unique *canonical switching*, which is in analogy to the unique way that a pair of adjacent independent letters in a string over an independence alphabet can be switched. In what sense switchings of pairs of direct derivation diagrams might *not* be canonical is illustrated by the following example.

Example 3.4 (On canonical switchings) Working in the category of $\mathbb{Sets}$, consider the "element deletion" rule $q_1 = \{\star\} \leftarrow\!\!\prec \varnothing \rightarrowtail \varnothing$ and its "conditional" variant $q_2 = \{1,2\} \leftarrow\!\!\supseteq\!\!\prec \{2\} \rightarrowtail \{2\}$. The application of the latter results in the deletion of the element onto which 1 is mapped in the set to be rewritten, but only if another element can be preserved meanwhile; the former rule does not have any such side condition and only deletes the element onto which $\star$ is mapped. Next, consider the following pair of sequential-independent direct derivation diagrams



where functions are described by the set of their input-output pairs with '$\mapsto$' as separation symbol.

The crucial point of this example is that – even though replacing the match function $\{1 \mapsto b, 2 \mapsto c\}$ by $\{1 \mapsto b, 2 \mapsto x\}$ does not lead to very drastic changes in the two direct derivation diagrams – the (corresponding) two ways to "reschedule" the pair of direct derivation diagrams, namely $\{a,b,c,d\} \models\!\!\Rightarrow_{\langle q_2, \{1\mapsto b, 2\mapsto c\}\rangle} \{a,c,x\} \models\!\!\Rightarrow_{\langle q_1, \{\star\mapsto a\}\rangle} \{c,x\}$ on the one hand and $\{a,b,c,d\} \models\!\!\Rightarrow_{\langle q_2, \{1\mapsto b, 2\mapsto x\}\rangle} \{a,c,x\} \models\!\!\Rightarrow_{\langle q_1, \{\star\mapsto a\}\rangle} \{c,x\}$ on the other hand do differ: the first one is *canonical* since it "directly" uses the independence pair to obtain the canonical switching whereas the other one uses the match $\{1 \mapsto b, 2 \mapsto x\}$, which involves the so far untouched element x.

The difference between the two switchings might become clearer by addition of a third direct derivation diagram which witnesses $\{c,x\} \models\!\!\Rightarrow_{\langle q_1, \{\star\mapsto x\}\rangle} \{c\}$,

weakly $\mathscr{M}$-adhesive category.

i.e. a second application of the rule q_1 at x. While the latter step is independent of the above two steps of the canonical switching, it would interfere with the application of q_2 at the match $\{1 \mapsto b, 2 \mapsto x\}$. ☙

This example might also be used to make clearer that, in general, the analogy between trace monoids and derivation up to concurrency is between pairs of independent *letters* on the one hand and pairs of sequential-independent applications of *rules* on the other hand. However, the involved morphisms of the complete derivation diagrams with their independence pairs do not have any (direct) counterpart in the world of traces. But already the fact that several different objects might occur in a sequence of direct derivations diagrams is in contrast to the single object of a trace monoid when the latter is seen as a one object category (cf. Example A.3).

The formal treatment of sequences of direct derivation diagrams up to concurrency starts with a generalization of strings (without any independence relation). The analogy is between strings over Σ, i.e. an elements in the free monoid Σ^*, on the one hand and arrows in the free category of the graph of the raw transformation system of a given a set of rules $\mathscr{R}$ in a category $\mathbb{C}$. This means that strings of letters are replaced by strings "compatible" direct derivation diagrams.

☼ Definition 3.5 (Category of DPO-derivations) Let $\mathbb{C}$ be a category and let $\mathscr{R}$ be a set of $\mathbb{C}$-spans, i.e. a set of rules. The DPO-*derivation graph of* $\mathscr{R}$ *in* $\mathbb{C}$ is the graph $G^{\mathbb{C}}_{\mathscr{R}} = \langle V, E, \mathrm{src}, \mathrm{tgt}\rangle$ that has all $\mathbb{C}$-objects as vertices and the edges between two nodes $A, B \in \mathrm{ob}(\mathbb{C})$ are all those DPO direct derivation diagrams that witness a derivation step $A \Rightarrow\!\!\!\!\!\!\cdot\;{}^{q}\Rightarrow B$ for some $q \in \mathscr{R}$. More formally $V = \mathrm{ob}(\mathbb{C})$, $E = \{\mathfrak{X} \mid \exists q \in \mathscr{R}.\, \exists A, B \in \mathrm{ob}(\mathbb{C}).\, \mathfrak{X} \text{ witnesses } A \Rightarrow\!\!\!\!\!\!\cdot\;{}^{q}\Rightarrow B\}$, and for $\mathfrak{X} \in E$ the equation $\langle \mathrm{src}(\mathfrak{X}), \mathrm{tgt}(\mathfrak{X})\rangle = \langle A, B\rangle$ holds if and only if $\mathfrak{X}$ witnesses that $A \Rightarrow\!\!\!\!\!\!\cdot\;{}^{q}\Rightarrow B$ holds for some $q \in \mathscr{R}$.

The DPO-*derivation category of* $\mathscr{R}$ *in* $\mathbb{C}$, denoted by $\mathrm{DPO}(\mathbb{C}, \mathscr{R})$, is the free category of the DPO-derivation graph $G^{\mathbb{C}}_{\mathscr{R}}$. A DPO $\mathscr{R}$*-derivation from* A *to* B is a morphism $\underset{\sim}{\mathfrak{X}}\colon A \to B$ in $\mathrm{DPO}(\mathbb{C}, \mathscr{R})$. ☼

The tilde is used to indicate that each morphism $\underset{\sim}{\mathfrak{X}}\colon A \to B$ in $\mathrm{DPO}(\mathbb{C}, \mathscr{R})$ is a (possibly empty) sequence of direct derivation diagrams $\mathfrak{X}_1 \ldots \mathfrak{X}_n$. Further, any non-identity arrow $\underset{\sim}{\mathfrak{X}}\colon A \to B$ in $\mathrm{DPO}(\mathbb{C}, \mathscr{R})$ is a direct derivation diagram if and only if $\underset{\sim}{\mathfrak{X}}$ does not have any non-trivial factorizations, which means that $\underset{\sim}{\mathfrak{X}}$ is a sequence of length one. The identity on $A \in \mathrm{DPO}(\mathbb{C}, \mathscr{R})$ is the empty sequence, and is written $\mathfrak{I}_A$.

Extending the notation for derivation steps, the expression '$\underset{\sim}{\mathfrak{X}}\colon A \Rightarrow\!\!\!\!\!\!\cdot\;{}^{\mathscr{R}}\Rightarrow B$' is another means to state that $\underset{\sim}{\mathfrak{X}}\colon A \to B$ is an arrow in $\mathrm{DPO}(\mathbb{C}, \mathscr{R})$, and

'$\mathcal{X}\colon A \models\!\!\Rightarrow_q B$' will be written to indicate that $\mathcal{X}$ is a DPO direct derivation diagram that witnesses a DPO rewriting step $A \models\!\!\Rightarrow_q B$. Moreover, the expression '$\mathcal{X}\mathcal{Y}\colon A \models\!\!\Rightarrow_q B \models\!\!\Rightarrow_p C$' is shorthand for '$\mathcal{X}\colon A \models\!\!\Rightarrow_q B$ and $\mathcal{Y}\colon B \models\!\!\Rightarrow_p C$'.

The central step in defining DPO-derivations up to concurrency is a proper definition of canonical switching that allows to relate parallel[34] derivation diagrams $\mathcal{X}, \mathcal{Z}\colon A \models\!\!\Rightarrow_{\mathscr{R}} B$ in the "same" way as independence relations relate pairs of letters in the case of trace monoids. The category of concurrent DPO-derivations or DPO-*traces* then arises by quotienting the DPO-derivation category by the '*is a canonical switching of*' relation.

3.2.1 Canonical switchings

To define the canonical switching of a pair of sequential-independent direct derivation diagrams in the style of Lemma 2.30, pairs of derivations must be switched by means of the constructions in the proof of sequential commutativity.

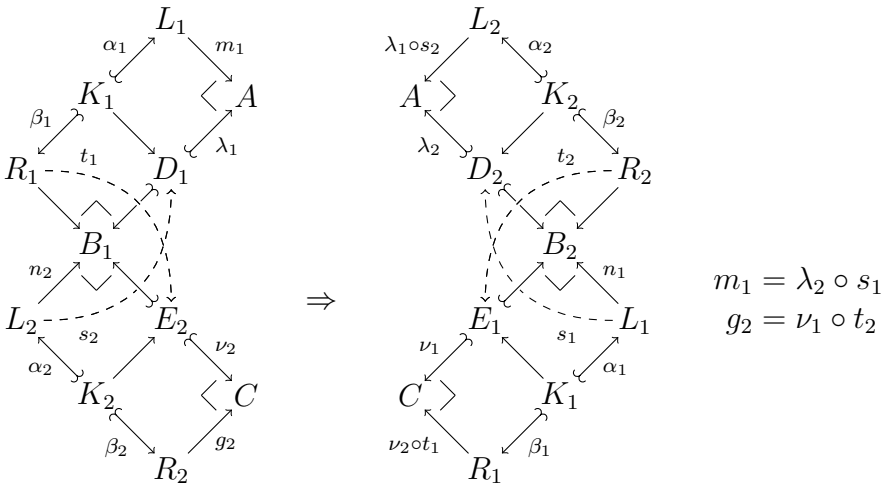


Figure 41: Switching of sequential-independent direct derivations

Proposition 3.6 (Sequential commutativity) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $q_1 = L_1 \leftarrow\!\alpha_1\!\rightarrowtail K_1 \rightarrowtail\!\beta_1\!\leftarrow R_1$ and $q_2 = L_2 \leftarrow\!\alpha_2\!\rightarrowtail K_2 \rightarrowtail\!\beta_2\!\leftarrow R_2$ be a pair of linear rules, and let the left diagram in Figure 41 comprise a

[34]The work 'parallel' here merely refers to the fact that the derivations are morphisms with the same domain and codomain. This is not to be confused with the notion of parallel derivations as the simultaneous execution of the two rewriting sequences.

pair of sequential-independent direct derivation diagrams that witness the two derivation steps $A \rightrightarrows^{\langle q_1, m_1 \rangle} B_1$ and $B_1 \rightrightarrows^{\langle q_2, n_2 \rangle} C$; further let $s_2 \colon L_2 \to D_1$ and $t_1 \colon R_1 \to E_2$ be the uniquely determined morphisms that make the diagram commute, i.e. let $\langle s_2, t_1 \rangle$ be an independence pair for the two direct derivation diagrams.

Then there exist a pair of sequential-independent direct derivation diagrams witnessing two derivation steps $A \rightrightarrows^{\langle q_2, \lambda_1 \circ s_2 \rangle} B_2 \rightrightarrows^{\langle q_1, n_1 \rangle} C$ as shown in the right hand diagram in Figure 41; in particular, the object B_2 is unique up to isomorphism.

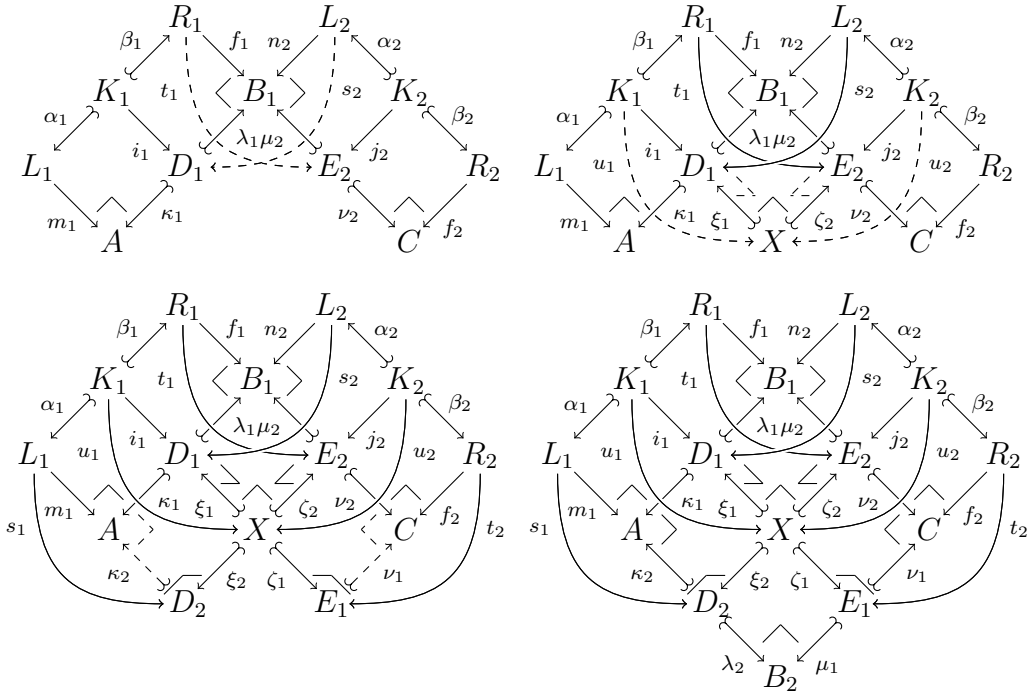


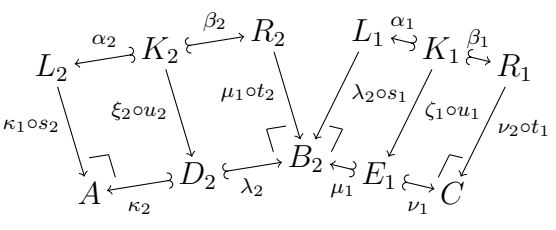
Figure 42: Proof outline of sequential commutativity

Proof. Based on the proof of Theorem 7.7 of [Lack and Sobociński, 2005], start with a pair of sequential-independent direct derivation diagrams as in the first diagram in Figure 42. Let $D_1 \leftarrow \xi_1 \rightarrowtail X \rightarrowtail \zeta_2 \rightarrow E_2$ be the pullback of $D_1 \rightarrowtail \lambda_1 \rightarrow B_1 \leftarrow \mu_2 \rightarrowtail E_2$, resulting in the pullback square $\begin{smallmatrix} & {\scriptstyle B_1} & \\ {\scriptstyle D_1} & \uparrow & {\scriptstyle E_2} \\ & X & \end{smallmatrix}$. Further, as illustrated in the second diagram in Figure 42, let $u_1 \colon K_1 \to X$ be the unique morphism satisfying $i_1 = \xi_1 \circ u_1$ and $t_1 \circ \beta_1 = \zeta_2 \circ u_1$, which exists because of the universal property of the latter pullback. Similarly, let $u_2 \colon K_2 \to X$

be the unique arrow which satisfies the two equations $j_2 = \zeta_2 \circ u_2$ and $s_2 \circ \alpha_2 = \xi_1 \circ u_2$. Using Lemma 7.8, this yields pushout squares ${}^{R_1}_{K_1}\uparrow^{\rightarrow}_{\rightarrow}\uparrow^{E_2}_{X}$ and ${}^{D_1}_{X}\uparrow^{\leftarrow}_{\leftarrow}\uparrow^{L_2}_{K_2}$, respectively, i.e. $R_1 - t_1 \rightarrow E_2 \leftarrow \zeta_2 \multimap X$ is the pushout of $R_1 \leftarrow \beta_1 \multimap K_1 - u_1 \rightarrow X$, and $L_2 - s_2 \rightarrow D_1 \leftarrow \xi_1 \multimap X$ is the pushout of $L_2 \leftarrow \alpha_2 \multimap K_2 - u_2 \rightarrow X$.

The next construction step is illustrated in the third diagram in Figure 42 and consists in taking pushouts $L_1 - s_1 \rightarrow D_2 \leftarrow \xi_2 \multimap X$ and $X \leftarrowtail \zeta_1 \rightarrow E_1 \leftarrow t_2 - R_2$ of $L_1 \leftarrow \alpha_1 \multimap K_1 - u_1 \rightarrow X$ and $X \leftarrow u_2 - K_2 \leftarrowtail \beta_2 \rightarrow R_2$, respectively, which yields the respective pushout squares ${}^{K_1}_{L_1}\downarrow^{\rightarrow}_{\rightarrow}\downarrow^{X}_{D_2}$ and ${}^{X}_{E_1}\downarrow^{\leftarrow}_{\leftarrow}\downarrow^{K_2}_{R_2}$. Now there exist unique morphisms $\kappa_2 \colon D_2 \hookleftarrow A$ and $\nu_1 \colon E_1 \hookleftarrow C$ which make the diagram commute, i.e. κ_2 satisfies the two equations $m_1 = \kappa_2 \circ s_1$ and $\kappa_1 \circ \xi_1 = \kappa_2 \circ \xi_2$, and for ν_1, both $f_2 = \nu_1 \circ t_2$ and $\nu_2 \circ \zeta_2 = \nu_1 \circ \zeta_1$ hold true. Moreover, using the Pushout Lemma, i.e. Lemma A.14, $D_1 \leftarrowtail \kappa_1 \rightarrow A \leftarrow \kappa_2 \multimap D_2$ is the pushout of $D_1 \leftarrow \xi_1 \multimap X \leftarrowtail \xi_2 \rightarrow D_2$ and similarly $E_2 \leftarrowtail \nu_2 \rightarrow C \leftarrow \nu_1 \multimap E_1$ is the pushout of $E_2 \leftarrow \zeta_2 \multimap X \leftarrowtail \zeta_1 \rightarrow E_1$, which results in pushout squares ${}^{D_1}_{A}\downarrow^{\leftarrow}_{\leftarrow}\downarrow^{X}_{D_2}$ and ${}^{X}_{E_1}\downarrow^{\rightarrow}_{\rightarrow}\downarrow^{E_2}_{C}$.

The final construction step is presented in the fourth diagram in Figure 42, in other words take a pushout $D_2 \leftarrowtail \lambda_2 \rightarrow B_2 \leftarrow \mu_1 \multimap E_1$ of the span $D_2 \leftarrow \xi_2 \multimap X \leftarrowtail \zeta_1 \rightarrow E_1$, which yields the pushout square ${}^{X}_{D_2}\downarrow^{\rightarrow}_{\rightarrow}\downarrow^{E_1}_{B_2}$. Using the Pushout Lemma once more, this results in the displayed pair of sequential-independent direct derivation diagrams. □



The main ingredient for the definition of canonical switchings are the four pullback squares in the center of the fourth diagram in Figure 42 which have X as pullback object together with the two morphisms $u_1 \colon K_1 \to X$ and $u_2 \colon K_2 \to X$. Though the idea of canonical switching might be considered natural the full details of an exhaustive definition are as follows.

☼ Definition 3.7 (Canonical switching couple) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $q_1 = L_1 \leftarrow \alpha_1 \multimap K_1 \leftarrowtail \beta_1 \rightarrow R_1$ and $q_2 = L_2 \leftarrow \alpha_2 \multimap K_2 \leftarrowtail \beta_2 \rightarrow R_2$ be a pair of linear rules in $\mathbb{C}$, and further let $\mathcal{X}_1\mathcal{Y}_2 \colon A \Rightarrow\!\!\!\!\!\!= q_1 \Rightarrow B_1 \Rightarrow\!\!\!\!\!\!= q_2 \Rightarrow C$ and $\mathcal{X}_2\mathcal{Y}_1 \colon A \Rightarrow\!\!\!\!\!\!= q_2 \Rightarrow B_2 \Rightarrow\!\!\!\!\!\!= q_1 \Rightarrow C$ be a pair of parallel derivations as shown in the thinly drawn part of Figure 43. A *canonical filler*[35] *between* $\mathcal{X}_1\mathcal{Y}_2$ *and* $\mathcal{X}_2\mathcal{Y}_1$ is a cospan $K_1 - u_1 \rightarrow X \leftarrow u_2 - K_2$ with a pair of spans $D_1 \leftarrowtail \xi_1 - X - \xi_2 \rightarrow D_2$ and $E_1 \leftarrowtail \zeta_1 - X - \zeta_2 \rightarrow E_2$ that satisfies the property that is illustrated in Figure 43, namely

[35] Apparently, this definition could be used to equip the derivation category with these fillers as cells, thus giving rise to a 2-category.

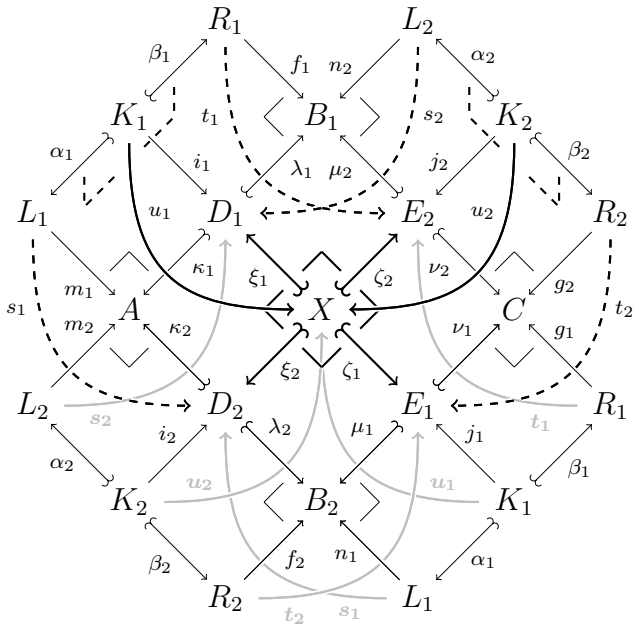


Figure 43: Filling the cell between canonical switchings

1. the pair of spans gives rise to the four pullback squares ${}^{D_1}_{A}\!\downarrow\!{}^{\leftarrow}_{\leftarrow}\!\downarrow{}^{X}_{D_2}$, ${}^{X}_{E_1}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{E_2}_{C}$, ${}^{B_1}_{D_1}\!\uparrow\!{}^{\leftarrow}_{\leftarrow}\!\uparrow{}^{E_2}_{X}$, and ${}^{D_2}_{X}\!\uparrow\!{}^{\rightarrow}_{\rightarrow}\!\uparrow{}^{E_1}_{B_2}$;

2. the cospan induces candidates for independence pairs based on arrows $s_1 \colon L_1 \to D_2$, $t_1 \colon R_1 \to E_2$, $s_2 \colon L_2 \to D_1$, and $t_2 \colon R_2 \to E_1$ such that

   (a) these arrows induce pullback complements, i.e. the composable pairs $\leftarrow_{s_1} - \leftarrow_{\alpha_1} -$, $\leftarrow_{t_1} - \leftarrow_{\beta_1} -$, $\leftarrow_{s_2} - \leftarrow_{\alpha_2} -$, and $\leftarrow_{t_2} - \leftarrow_{\beta_2} -$ are pullback complements of $\leftarrow_{\xi_2} - \leftarrow_{u_1} -$, $\leftarrow_{\zeta_2} - \leftarrow_{u_1} -$, $\leftarrow_{\xi_1} - \leftarrow_{u_2} -$, $\leftarrow_{\zeta_1} - \leftarrow_{u_2} -$, respectively, and

   (b) the whole diagram commutes, i.e.

   - $m_1 = \kappa_2 \circ s_1$, $i_1 = \xi_1 \circ u_1$, $f_1 = \mu_2 \circ t_1$,
   - $n_2 = \lambda_1 \circ s_2$, $j_2 = \zeta_2 \circ u_2$, $g_2 = \nu_1 \circ t_2$,
   - $m_2 = \kappa_1 \circ s_2$, $i_2 = \xi_2 \circ u_2$, $f_2 = \mu_1 \circ t_2$, and
   - $n_1 = \lambda_2 \circ s_1$, $j_1 = \zeta_1 \circ u_1$, $g_1 = \nu_2 \circ t_1$.

Finally, the two derivations $\mathcal{X}_1\mathcal{Y}_2$ and $\mathcal{X}_2\mathcal{Y}_1$ form a *canonical switching couple* if there is a canonical filler between $\mathcal{X}_1\mathcal{Y}_2$ and $\mathcal{X}_2\mathcal{Y}_1$, and in this situation $\mathcal{X}_2\mathcal{Y}_1$ is said to be a *canonical switching* of $\mathcal{X}_1\mathcal{Y}_2$. ☼

Remark 3.8 (On canonical switchings) The choice of a canonical switching couple for two pairs of sequential-independent derivation diagrams is clearly only unique up to a compatible isomorphism at X. However, the issues of a formalization of the simple idea of the switching of a pair of sequential-independent direct derivations are subtle.

For illustrative purposes, take the following example of a non-canonical switching. Take the set $A = \{\langle 0, a\rangle, \langle 0, b\rangle\} \cup \{\langle 1, a\rangle, \langle 1, b\rangle\}$, i.e. A is of the form $A = A' + A'$. Moreover, take two sequential-independent applications of the rule $\{\star\} \leftarrow \varnothing \rightarrow \varnothing$ with matches $m_1 = \{\star \mapsto \langle 0, a\rangle\}$ and $m_2 = \{\star \mapsto \langle 0, b\rangle\}$. A non-canonical switching of these two rule applications arises via the matches $m_1' = \{\star \mapsto \langle 1, b\rangle\}$ and $m_2' = \{\star \mapsto \langle 1, a\rangle\}$, which act on the second copy of $\{a, b\}$ instead of the first one.

3.2.2 Traces of derivations

Having the notion of canonical switching couple at hand, the category of DPO-traces for a given set of rules arises from the derivation category by identifying all parallel morphisms that arise from each other by repeatedly replacing pairs of sequential-independent direct derivation diagram by (one of) their canonical switchings. Using the construction of quotient categories (see [Mac Lane, 1998, page 51f]), the resulting category of DPO-*traces* can be defined as follows.

Definition 3.9 (Category of DPO-traces) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category and let $\mathscr{R}$ be a set of linear rules; further, assign to each pair of objects $A, C \in \mathbb{C}$ the set $J_{A,C} \subseteq \mathrm{DPO}(\mathbb{C}, \mathscr{R})(A, C)$ that contains all canonical switching couples consisting of pairs of direct derivation diagrams leading from the object A to C, i.e.

$$J_{A,C} = \left\{ \langle \mathcal{X}_1\mathcal{Y}_2, \mathcal{Y}_1\mathcal{X}_2\rangle \;\middle|\; \begin{array}{l} \mathcal{X}_1\mathcal{Y}_2, \mathcal{Y}_1\mathcal{X}_2 \colon A \to C \text{ in } \mathrm{DPO}(\mathbb{C}, \mathscr{R}) \\ \mathcal{Y}_1\mathcal{X}_2 \text{ is a canonical switching of } \mathcal{X}_1\mathcal{Y}_2 \end{array} \right\}.$$

Then the *category of* DPO-$\mathscr{R}$*-traces*, written $\mathrm{DPO}(\mathbb{C}, \mathscr{R})^{\sim}$, is the quotient category $\mathrm{DPO}(\mathbb{C}, \mathscr{R})/\!\!/J$, which is the category obtained from the generators $G^{\mathbb{C}}_{\mathscr{R}}$ and the family of relations J (see [Mac Lane, 1998, page 51f] for more details). Finally, a DPO-$\mathscr{R}$*-trace* is defined as a morphism in the category $\mathrm{DPO}(\mathbb{C}, \mathscr{R})^{\sim}$.

This definition is tailored to emphasize the similarities between independence relations in the theory of Mazurkiewicz traces, and sequential independence of pairs of direct derivation diagrams. Alternatively, a more direct

definition of switch-equivalence of pairs of derivations with the same domain and codomain is the following one (cf. [Baldan et al., 2006a]).

Definition 3.10 (Switch-equivalence) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category let $\mathscr{R}$ be a set of linear rules. Further let $\underset{\sim}{\mathcal{X}}, \underset{\sim}{\mathcal{Z}}\colon A \models\!\!\mathscr{R}\!\!\Rightarrow B$ be two derivations in $\text{DPO}(\mathbb{C}, \mathscr{R})$. Then $\underset{\sim}{\mathcal{X}}$ and $\underset{\sim}{\mathcal{Z}}$ are *direct switchings of each other*, written $\underset{\sim}{\mathcal{X}} \sim_{\text{sw}} \underset{\sim}{\mathcal{Z}}$, if there exists a canonical switching couple $\langle \mathcal{W}_1\mathcal{V}_2, \mathcal{V}_2\mathcal{W}_1 \rangle$ and a factorization $\underset{\sim}{\mathcal{X}} = \underset{\sim}{\mathcal{X}}' \mathbin{⨟} \mathcal{W}_1 \mathbin{⨟} \mathcal{V}_2 \mathbin{⨟} \underset{\sim}{\mathcal{X}}''$ of $\underset{\sim}{\mathcal{X}}$ such that $\underset{\sim}{\mathcal{Z}} = \underset{\sim}{\mathcal{X}}' \mathbin{⨟} \mathcal{V}_2 \mathbin{⨟} \mathcal{W}_1 \mathbin{⨟} \underset{\sim}{\mathcal{X}}''$. Then *switch-equivalence*, denoted by $\approx_{\text{sw}}$, is the reflexive-transitive closure of the direct switching relation $\sim_{\text{sw}}$, in signs $\approx_{\text{sw}} := (\sim_{\text{sw}})^*$.

It is immediate that two derivations are switch-equivalent if and only if they represent the same morphism in the category of DPO-traces.

Lemma 3.11 (Switch-equivalence) Switch-equivalence is an equivalence.

Proof idea. The desired follows from the observation that direct switch equivalence is symmetric. □

This concludes the discussion on switch equivalence and traces of derivations with linear rules and arbitrary matches. With minor complications a similar notion of traces with $\mathscr{M}$-matches and arbitrary left-linear rules could be given. Also the theory of parallel direct derivations as presented in [Bonchi and Heindel, 2006] readily lifts to weakly $\mathscr{M}$-adhesive categories (with $\mathscr{M}$-effective unions). This would allow to execute pairs of sequential-independent rule applications in one step by combining two rules into a (usually larger) parallel rule.

Moreover, with the latter provisions, it should also be possible to lift the ideas of [Kreowski, 1986] to the abstract level of adhesive rewriting system to obtain so-called *canonical* derivations. The latter are obtained from a given derivation by successively substituting sequential-independent rule applications by parallel, simultaneous rule applications until a “maximally parallelized” derivation is obtained. However, the technical details of canonical derivations are complex and thus one might be interested in (more elegant) alternative approaches.

One alternative approach is based on the notion of (deterministic) process as known from the theory of Petri nets [Goltz and Reisig, 1983]. The latter notion has been generalized to graph transformation systems [Montanari et al., 1996], and also applies to adhesive rewriting systems [Baldan et al., 2006a] as described in the next section.

Causality and conflict for processes of derivations

4

A finite computation in a system consists of a start state, a (possibly empty) sequence of transitions or *events*, and an end state.[36] Using one of the pushout approaches, each event corresponds to a rewriting step, and the whole computation is captured by a morphism in the derivation category, i.e. a sequence of matching direct derivation diagrams. To account for concurrency in systems, as described in Section 3.2, derivations can be quotiented using the notion of sequential independence, and the resulting concept of trace captures the idea of a computation up to concurrency.

This section will be concerned with an alternative approach to account for concurrency. It is based on causal dependencies and conflicts between events in systems instead of independence of events. The main idea might be described as follows: instead of beginning with the set of all possible "manifestations" of a computation up to concurrency, which then is quotiented according to independence, the causal approach starts from a partial order on the events in a computation such that the set actual "manifestations" correspond to linearizations of this partial order. It will turn out that pushout rewriting models of systems allow to relate the events that occur in finite computations of a system by a partial order which models causality in the system in a suitable way. Generalizing the results for Petri nets and graph transformations of [Baldan, 2000], it will turn out that this causality based approach is equivalent to the one based on sequential independence and switch equivalence.

That the use of partial orders is a powerful means to avoid unnecessarily large representations of systems, i.e. that partial orders allow to avoid the so-called *state-space explosion*, is a well-know fact from the theory of Petri nets (see e.g. [McMillan, 1995]). Hence one motivation to study partial order based semantics of double pushout rewriting is the goal of obtaining a small representations of system runs by exploiting the concurrency that is already present in the system.

Moreover, if a given system comes equipped with some (informal) description of causality, an alternative, causality based presentation of traces might be more suitable for practical applications. The reason for this is that in such a situation, it is probably easier to check whether all causal dependencies are

[36] In this thesis, only *closed* systems are considered, i.e. systems are "isolated" insofar as they do not interact with any environment; the opposite class of systems are the open ones. There is no connection to the open and closed sets from topology.

present in the model of the given system, since not only the system but also its model comes equipped with a notion of causality. In this way, the task to develop sound models of systems might become less error prone (the same argument also holds for the absence of spurious dependencies in the model). Further, also in cases where the causal dependencies in the modeled system only have a rough informal or incomplete description, formal models might help to arrive at a better understanding of causality in the system itself, which in turn could improve the chance to find and analyze errors.

To illustrate the central idea of causality based semantics of DPO-traces, recall that DPO-traces can be conceived of as generalizations of trace monoids, which can be faithfully represented using pomsets[37]. In analogy to these pomset representations, this section develops causality based description of DPO-computation up to concurrency. The result are so-called *processes* of derivations as developed in the author's [Baldan et al., 2006a]. In comparison with DPO-traces, which are essentially switch-equivalence classes of derivations, processes of derivations will turn out to be relatively concrete models of finite computations up to concurrency.

Overview of the section. The main goal of this section is to present (a generalization of) the results of the author's [Baldan et al., 2006a]. This is done by giving an analogy with the bijective correspondence between Mazurkiewicz traces and partially ordered multisets (pomsets), which are isomorphism classes of labelled partial orders. The guiding idea is that labelled partial orders are to Mazurkiewicz traces what *trace processes* will be to DPO-traces. The prerequisites from [Diekert and Métivier, 1997] are reviewed in Section 4.1 and are also illustrated with examples.

The development of the theory starts in Section 4.2, which introduces the notion of oriented transformation system (OTS) as a rough analogon to labelled partial orders. The first observation is that any double pushout derivation can be represented as an OTS which is constructed in analogy to an algorithm in [Diekert and Métivier, 1997] which computes labelled partial orders for strings over an independence alphabet; those oriented transformation systems that arise in this manner are called *trace processes*. The main result of the present section is Theorem 4.14, which states that the underlying derivation of a trace processes can be recovered up to switch-equivalence (and isomorphism)

[37] A pomset is a *partially ordered multiset* as formalized in Definition 4.2. The word *pomset* is shorthand for partially ordered multiset and arises from the latter phrase in the same way as *poset* arose from partially ordered set.

by *execution*, which is the counterpart of the linearization of a labelled partial order. Moreover, switch-equivalent derivations give rise to isomorphic trace processes in the same way as two representatives of a Mazurkiewicz trace yield isomorphic labelled partial orders, i.e. the same pomset.

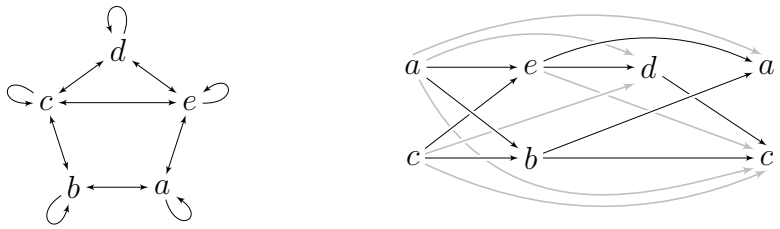
Finally, Section 4.3 makes the conceptual connection to the theory of Petri nets and graph transformation. It gives a static characterization of (isomorphism classes of) oriented transformation systems which arise from DPO-derivations by means of the algorithmic construction in Section 4.2. The resulting notion is dubbed *concatenable* DPO*-process* since it is a generalization of previous notions of concatenable processes in the area of graph transformation [Montanari et al., 1996] and Petri nets (see e.g. [Sassone, 1996]).

4.1 Reviewing pomsets of Mazurkiewicz traces

Before starting the actual development for double pushout rewriting in weakly adhesive categories, the relevant facts about labeled partial orders and pomsets are reviewed. Recall that traces are words over some alphabet Σ quotiented by an equivalence that is generated by some symmetric, irreflexive relation $I \subseteq \Sigma \times \Sigma$, which is called an *independence relation*; further this relation gives rise to the *independence alphabet* $\langle \Sigma, I \rangle$ (see [Diekert and Métivier, 1997]). The complement $D = \Sigma \times \Sigma \setminus I$ of an independence relation I is a *dependence relation*, and the pair $\langle \Sigma, D \rangle$ a *dependence alphabet*. Each (in)dependence relation can be represented by a (simple) graph that has Σ as the set of nodes. Given an independence alphabet $\langle \Sigma, I \rangle$, each word $w \in \Sigma^*$ gives rise to a trace $[w]_I$ which consists of all words that can be obtained from w by repeatedly switching neighboring letters that are related by I. Before giving the precise algorithm to generate the pomset representation of such a trace $[w]_I$, consider the following example from [Diekert and Métivier, 1997].

Example 4.1 (Pomset representation of a trace) Let $\Sigma = \{a, b, c, d, e\}$ be an alphabet, and let $I = \{\langle a, d\rangle, \langle d, a\rangle, \langle a, c\rangle, \langle c, a\rangle, \langle d, b\rangle, \langle b, d\rangle, \langle e, b\rangle, \langle b, e\rangle\}$ be an independence relation; the latter induces the dependence relation $D = \Sigma \times \Sigma \setminus I$ as illustrated in Figure 44(a).

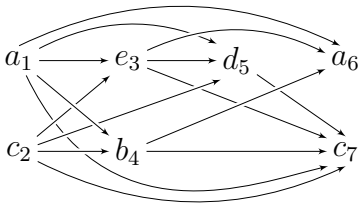
Further, Figure 44(b) displays the *pomset* of the trace represented by the word $acebdac \in \Sigma^*$ in the context of the independence relation I. Nodes correspond to letter occurrences, and an edge from one node to another one expresses that the source node must necessarily precede the target node in any representative of the trace $[acebdac]_I$. One might want to think of these edges as representing the causal dependencies between the events of some



(a) Dependence relation (b) Pomset representation of $[acebdac]$

Figure 44: Illustration of the pomset representation of a trace

computation in a system – under this interpretation, each letter in the alphabet corresponds to a certain type of event.



More detailed, consider the "more concrete" graph displayed to the right. Also this graphical illustration contains all *occurrences* of letters in the word *acebdac* as nodes. It is a Σ-node-labeled graph with $\{a_1, c_2, e_3, b_4, d_5, a_6, c_7\}$ as the set of nodes where the subscripts indicate the (accidental) positions of the letter occurrences in the word *acebdac*; the label of each node is obtained by removing the subscript. Further, in this graph, there is an edge from x_i to y_j if and only if both $i < j$ and $\langle x, y\rangle \in D$ hold. The corresponding *pomset* in Figure 44(b) is obtained from this graph to the right by "hiding" the arbitrary choice of the set of nodes, thus obtaining a multiset over Σ (with a partial order).

As a remark, the gray edges in Figure 44(b) could be removed and then recovered from the transitive closure of the relation described by the black edges, which is the *Hasse diagram* of the poset used in the pomset representation of the trace $[acebdac]_I$.

Next, an algorithm to obtain the pomset of a trace will be described; later, the idea of a process (of a derivation) will arise as a generalization of the concept of a pomset (of a trace). The construction of the pomset from a given word (which represents a trace) in the context of an independence alphabet is a recursive process: it starts with the empty pomset; then, as long as there are letters left, a new node is added to the pomset for the next letter (occurrence) and the partial order which describes the dependencies between letter occurrences is adjusted accordingly.

Formally, as detailed in Definition 4.2 below, a pomset is a *labeled partial order* (LPO) taken up to isomorphism – thus hiding the arbitrary choice of the elements of the partial order. Based on this concept, Algorithm 1 (which is based again on [Diekert and Métivier, 1997]) makes the idea of the last paragraph more precise: given a string $a_1 \cdots a_n \in \Sigma^*$ with a dependence relation $D \subseteq \Sigma \times \Sigma$, the function STRINGTOLPO$(\Sigma, D, a_1 \cdots a_n)$ computes (a representative of) the pomset of the trace $[a_1 \cdots a_n]$, which is the trace represented by $a_1 \cdots a_n$ in the context of the dependency relation $D \subseteq \Sigma \times \Sigma$.

Algorithm 1 The *string-to*-LPO algorithm using a dependency $D \subseteq \Sigma \times \Sigma$

function STRINGTOLPO$(\Sigma,\ D \subseteq \Sigma \times \Sigma,\ a_1 \cdots a_n \in \Sigma^*)$
  $V := \varnothing;\ \prec := \varnothing$ ▷ empty Hasse diagram $\prec \subseteq V \times V$ (no nodes yet)
  $\lambda := \varnothing$ ▷ the labeling of nodes
  $Min_\prec := \varnothing$ ▷ the set of $\prec$-minimal elements of V
  $i := n$ ▷ index of last letter a_n
  while $i > 0$ **do**
    $V := V \uplus \{v_i\};\ \lambda(v_i) := a_i$ ▷ new a_i-labeled node
    $Min_\prec := Min_\prec \uplus \{v_i\}$ ▷ v_i minimal (right now)
    for all $w \in Min_\prec \setminus \{v_i\}$ **do** ▷ for all (other) minimal nodes
      if $(a_i, \lambda(v)) \in D$ **then** ▷ check whether w depends on v_i
        $\prec := \prec \uplus \{\langle v_i, w\rangle\}$ ▷ update $\prec$
        $Min_\prec := Min_\prec \setminus \{w\}$ ▷ update $Min_\prec$
      end if
    end for
    $i := i - 1$
  end while
  return $\langle V, \prec^*, \Sigma, \lambda\rangle$ ▷ reflexive-transitive closure $\prec^*$
end function

☼ DEFINITION 4.2 (Labelled partial order and pomset) A Σ-*labeled partial order* (LPO) is a quadruple $\langle V, \preceq, \Sigma, \lambda\rangle$ where V is a set, $\preceq \subseteq V \times V$ is a partial order, Σ is a set of *labels*, and $\lambda\colon V \to \Sigma$ is a *labeling function*. An LPO-*morphism* between Σ-labeled LPOs $\langle V_1, \preceq_1, \Sigma, \lambda_1\rangle$ and $\langle V_2, \preceq_2, \Sigma, \lambda_2\rangle$ is a function $f\colon V_1 \to V_2$ that is compatible with the labels and the partial orders, i.e. such that both the equation $\lambda_2 \circ f(u) = \lambda_1(u)$ and the implication $u \preceq_1 v \Rightarrow f(u) \preceq_2 f(v)$ hold for all $u, v \in V_1$. This gives rise to a category of labeled partial orders, having identity functions and function composition as identities and composition, respectively.

A *partially ordered multiset* or *pomset* is an isomorphism class of labeled partial orders (where an isomorphism in the category of LPOs turns out to be a bijective LPO-morphism $i\colon V_1 \to V_2$ that satisfies the bi-implication $u \preceq_1 v \Leftrightarrow i(u) \preceq_2 i(v)$ for all $u, v \in V_1$). ☼

As [Diekert and Métivier, 1997] write, "it is sometimes convenient to identify a trace [...] with its induced labeled partial order" – and be it just because of their graphical representation, which usually conveys the contained information in an easily accessible way. Though they ignore the structure of states, pomsets are the paradigm for the concurrent semantics of pushout based rewriting and will appear once more below in Section 4.3, where the focus is changed towards the static analysis of systems, i.e. to those techniques that do not need to execute computations of a given system to check properties and hence are usually more efficient. There the following characterization of pomsets as certain finite and *acyclic* Σ-node-labeled graphs will be convenient to explain the most central ideas of the causality based concurrent semantics of computations modeled using single and double pushout rewriting.

☼ Definition 4.3 (Dependence graph [Diekert and Métivier, 1997]) A Σ-*node-labeled graph* is a triple $\langle V, E, \lambda\rangle$ such that $\langle V, E\rangle$ is a simple graph, and $\lambda\colon V \to \Sigma$ is a labeling function. A morphism between Σ-node-labeled graphs $\langle V_1, E_1, \lambda_1\rangle$ and $\langle V_2, E_2, \lambda_2\rangle$ is a graph morphism $\varphi\colon \langle V_1, E_1\rangle \to \langle V_2, E_2\rangle$ that is compatible with the labellings, i.e. such that $\lambda_1 = \lambda_2 \circ \varphi$.

Let $D \subseteq \Sigma \times \Sigma$ be a dependence relation, i.e. a symmetric, reflexive relation. A $\langle \Sigma, D\rangle$-*dependence graph* is an isomorphism class of a Σ-node-labeled acyclic graphs $\langle V, E, \lambda\rangle$, written $[\langle V, E, \lambda\rangle]$, such that

- the graph $\langle V, E\rangle$ is acyclic, i.e. $E \subseteq V \times V$ is an acyclic relation;
- the set of nodes V is at most countable;
- the induced partial order $E^* \subseteq V \times V$ is well-founded[38];
- the labeling $\lambda\colon V \to \Sigma$ respects the dependence relation $D \subseteq \Sigma \times \Sigma$, i.e. $\langle \lambda(u), \lambda(v)\rangle \in D$ if and only if $\langle u, v\rangle \in (E \cup E^{-1} \cup \mathrm{id}_V)$.

☼

Pomsets and traces are two sides of the same coin. Hence also pomsets form a monoid which inherit their composition from the represented traces.

[38] A partial order $E \subseteq V \times V$ is *well-founded* if it has no infinitely decreasing chains $\cdots\, w\ E\ v\ E\ u$, i.e. if every non-empty subset $\varnothing \neq V' \subseteq V$ has a *minimal element*, which is an element $v_0 \in V'$ such that $\{v' \in V' \mid v'\ E\ v_0\} = \{v_0\}$.

However, this composition operation can be described directly on the whole set of dependence graphs, thus arriving at the monoid of dependence graphs, written $\mathbb{G}(\Sigma, D)$, where Σ is an alphabet and D is a dependence relation. The following definition is again based on [Diekert and Métivier, 1997].

☼ Definition 4.4 (Monoid of dependence graphs) Let Σ be an alphabet, and let $D \subseteq \Sigma \times \Sigma$ be a dependence relation. The *monoid of dependence graphs over* $\langle \Sigma, D \rangle$, written $\mathbb{G}(\Sigma, D)$, has dependence graphs as elements, the neutral element is $[\langle \varnothing, \varnothing, ¡ \colon \varnothing \to \Sigma \rangle]$ where $¡ \colon \varnothing \to \Sigma$ is the unique labeling of the empty set, and the composition of two dependence graphs $[\langle V_1, E_1, \lambda_1 \rangle]$ and $[\langle V_2, E_2, \lambda_2 \rangle]$ is their union[39] with additional edges that relate nodes of the first graph with those of the second one whenever this is required according to D, namely $[\langle V_1, E_1, \lambda_1 \rangle] \cdot [\langle V_2, E_2, \lambda_2 \rangle] = [\langle V', E', \lambda' \rangle]$ where $V' = V_1 \cup V_2$, $\lambda' = \lambda_1 \cup \lambda_2$ and $E' = (E_1 \cup E_2) \cup \{\langle v_1, v_2 \rangle \mid \langle \lambda_1(v_1), \lambda_2(v_2) \rangle \in D\}$.[40] ☼

4.2 Generalizing pomsets to processes of DPO traces

Before addressing the generalization of the actual construction principle for labeled partial orders and pomsets of traces to *processes* of DPO-traces – again, these *processes* will be constructed as a "stack" of rule occurrences of a derivation representative of a given DPO-trace – it is necessary to provide a suitable general framework that allows to generalize the idea to represent trace monoids via pomsets to obtain an analogous representation of the category of DPO traces via processes.

This preliminary problem of ascending from trace monoids to categories of DPO-traces is connected with the phenomenon that each trace monoid essentially *is* a one object category, and hence it is always clear "where" pomsets of traces must "start" and "end", namely "at" the unique object. The following example is an attempt to illustrate this phenomenon by an encoding of the (in)dependence of pairs of letters in traces by means of suitable rules which act on the dependence relation.

☙ Example 4.5 (Traces into graphs) Consider the alphabet $\Sigma = \{a, b, c, d\}$ with the independence relation $I = \{\langle a, b \rangle, \langle b, a \rangle, \langle b, c \rangle, \langle c, b \rangle, \langle c, d \rangle, \langle d, c \rangle\}$, thus obtaining an independence alphabet $\langle \Sigma, I \rangle$, which is just the graph $\textcircled{a} \leftrightarrow \textcircled{b} \leftrightarrow \textcircled{c} \leftrightarrow \textcircled{d}$; the corresponding dependence relation D can be defined pic-

[39] The involved sets can all be assumed to be disjoint, i.e. $V_1 \cap V_2 = \varnothing$ and $E_1 \cap E_2 = \varnothing$ without loss of generality.

[40] Here $\lambda_1 \cup \lambda_2$ is the function $\lambda' \colon V_1 \cup V_2 \to \Sigma$ which maps all $v_1 \in V_1$ to $\lambda_1(v_1) \in \Sigma$ and all $v_2 \in V_2$ to $\lambda_2(v_2) \in \Sigma$.

torially by $b\,d\,a\,c$, i.e. the dependency relation is $D = (\Sigma \times \Sigma \setminus I) = \{\langle b,d\rangle, \langle d,b\rangle, \langle d,a\rangle, \langle a,d\rangle, \langle c,a\rangle, \langle a,c\rangle\} \cup \{\langle a,a\rangle, \langle b,b\rangle, \langle c,c\rangle, \langle d,d\rangle\}$.

Postponing the details of the category that is used for rewriting to the end of this example, define rules for each letter in the dependence alphabet $\langle \Sigma, D\rangle$ as follows: the rule for the letter b is $q_b := b\,d \leftarrowtail b\ d \rightarrowtail b\,d$, and similarly the one for c is $q_c := a\,c \leftarrowtail a\ c \rightarrowtail a\,c$. This is to the effect that both q_b and q_c can be applied at the graph $b\,d\,a\,c$ independently of each other at the obvious matches (either sequentially or in parallel): the first rule is operating at the left end of the graph while the latter one is manipulating the right end. In contrast, the rule for d, namely $q_d := b\,d\,a \leftarrowtail b\ d\ a \rightarrowtail b\,d\,a$ is dependent on q_b (since both remove the edge between b and d) but independent of q_c (because the components of q_c and q_d do not have any edges in common). Similarly, the final rule $q_a := d\,a\,c \leftarrowtail d\ a\ c \rightarrowtail d\,a\,c$ is dependent on q_d and q_c, but independent of q_b (see Figure 45 for a summary of the rule set $\{q_a, q_b, q_c, q_d\}$).

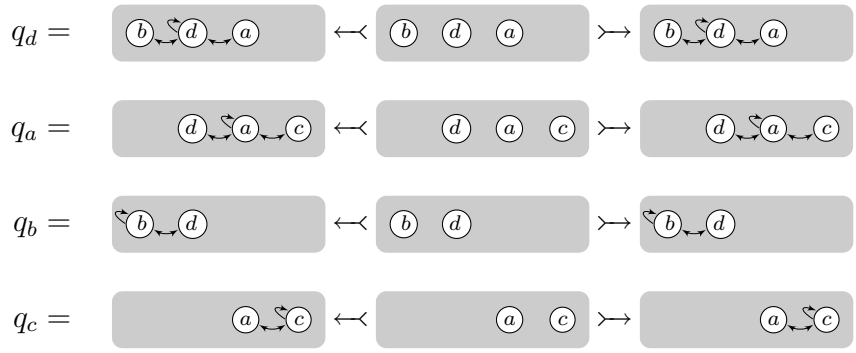


Figure 45: Encoding dependence alphabets using graph rules

Indeed, $\langle x, y\rangle \in D$ if and only if q_x is causally dependent on q_y in the sense of Definition 4.15 below; similarly, $\langle x, y\rangle \in I$ are independent of each other if and only if q_x and q_y are not causally related. In particular, each rule is (causally) dependent on itself.

This correspondence can be extended to arbitrary dependence alphabets $\langle \Sigma, D\rangle$ by constructing a set of rules $Q_{\langle \Sigma, D\rangle} = \{q_a \mid a \in \Sigma\}$ along the lines of this example. More precisely, for each $a \in \Sigma$, the node set of each rule component of q_a contains all nodes adjacent to a, and the left and right hand sides also contain all witnessing edges and the loop at a. As a result, application of the rule q_a at $\langle \Sigma, D\rangle$ using the obvious match amounts to deleting and recreating all edges that are incident to the node a in the graph $\langle \Sigma, D\rangle$.

Now, having defined such a set of rules $Q_{\langle\Sigma,D\rangle}$, e.g. $\{q_a, q_b, q_c, q_d\}$ in the present concrete case, sequences of letters in Σ are in one-one correspondence with derivations using rules in $Q_{\langle\Sigma,D\rangle}$, and similarly DPO-traces arise as counterparts to elements of the trace monoid induced by $\langle\Sigma, D\rangle$. For example abc corresponds to the derivation $\langle\Sigma, D\rangle \vDash_{q_a}\Rightarrow \langle\Sigma, D\rangle \vDash_{q_b}\Rightarrow \langle\Sigma, D\rangle \vDash_{q_c}\Rightarrow \langle\Sigma, D\rangle$ using the obvious matches, and similarly the Mazurkiewicz trace $[abc]_I$ is associated with the switch-equivalence class represented by this derivation.

This leads back to the main objective of this example, namely the illustration of the idea, that the single object of a trace monoid (considered as a one object category) "at" which rewriting takes place is implicit and does not change. In the context of this example, one might want to think of the graph $\bullet_b \to \bullet_d \to \bullet_a \to \bullet_c$ as *the*[41] object at which rewriting takes place: all derivations using rules in Q start and end at the latter graph.[42] This is in stark contrast to the typical examples of double pushout derivations, in which source and target of DPO-derivations are not isomorphic to each other, which reflects the fact that the structure of systems may change over time.

Finally, to make this example a proper instance of double pushout rewriting in weakly ($\mathscr{M}$-)adhesive categories and to make the notion of dependency between rules more precise, the choice for the ambient category is the slice category[43] $\mathtt{Graphs}\downarrow\langle\Sigma, D\rangle$, which is weakly adhesive; the objects of the latter category are graphs *typed over* $\langle\Sigma, D\rangle$,[44] i.e. graphs that come equipped with a morphism into $\langle\Sigma, D\rangle$. In the case of the present example, this is the category $\mathtt{Graphs}\downarrow \bullet_b \to \bullet_d \to \bullet_a \to \bullet_c$.

This is to the effect that also the rules in $Q_{\langle\Sigma,D\}}$ are rules in $\mathtt{Graphs}\downarrow\langle\Sigma, D\rangle$, and the dependency between rules can be defined using the fact that the components of the rules are actually morphisms into the graph $\langle\Sigma, D\rangle$, which happen to be monic, and they are assumed to be inclusions for the sake of simplicity. Hence all rule components are actually subgraphs of $\langle\Sigma, D\rangle$.

[41] Here, some technical issues of isomorphisms are ignored. However, this "nuisance" of *isomorphism up to* can be eliminated in general by application of the axiom of choice, which makes it possible to assume w.l.o.g. that *isomorphic* implies *equal*; this is achieved by means of a suitable choice of skeletons of categories.

[42] Further, it becomes apparent once more that the crucial information is contained in the direct derivation diagrams of derivations, since the rewriting relation in this example case is essentially the identity on $\bullet_b \to \bullet_d \to \bullet_a \to \bullet_c$.

[43] The notion of slice category is given in Definition A.25, which also introduces some notational conventions.

[44] Here, the simple graph $\langle\Sigma, D\rangle$ is considered as an object of $\mathtt{Graphs}$ using the usual embedding of the category of simple graphs $\mathtt{sGraphs}$ into the category of (multi-)graphs $\mathtt{Graphs}$.

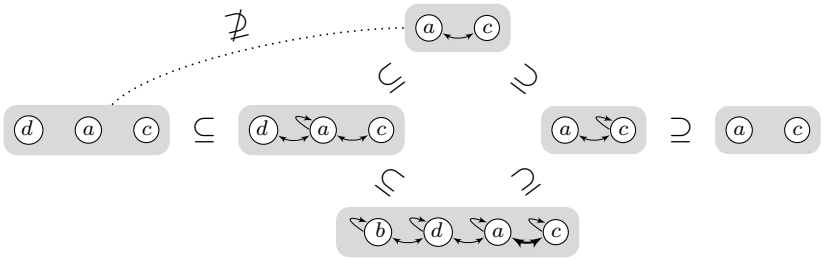


Figure 46: Causal dependence between q_a and q_c of Example 4.5

Finally, a rule q_x is (causally) dependent on another one q_y if the intersection of the left hand side of q_x with the right hand side of q_y contains an edge (which then is not contained in the gluing graph of q_y – see also Definition 4.15 below); in particular each rule q_x is dependent on itself because of the loop at x. For the two rules q_c and q_a, Figure 46 illustrates the reason why q_c is causally dependent on q_a.

Summarizing this example, the crucial difference between derivations and DPO-traces using a given set of rules in comparison with words and Mazurkiewicz traces over an independence alphabet consists in the fact that they define transformations from their source to their target objects (while carrying additional information concerning intermediate steps and concurrency or dependence of the involved rewriting steps); in particular the source and the target objects may differ (though they might happen to be the same). As a secondary feature, Example 4.5 shows how slice categories can be used to encode dependencies between rules.

As for the first point – regardless of issues of concurrency and causality – the counterparts of pomsets of traces need to carry information about the "outer appearance" of a derivation in the same way as every derivation in the trace category $\mathrm{DPO}(\mathbb{C})^{\sim}$ has a source and a target. Hence, in direct analogy to $\mathrm{DPO}(\mathbb{C})^{\sim}$, also the *trace processes* defined below in Definition 4.8 will have a source and target object.

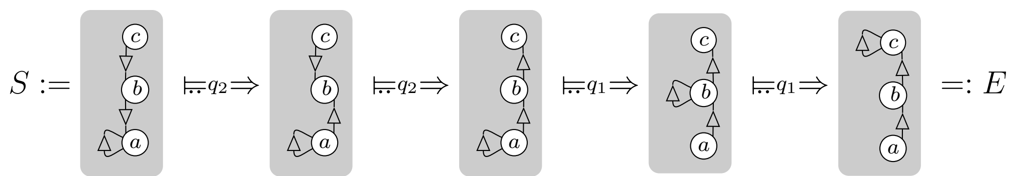
As for the second point, the dependence relation between rule occurrences can be given implicitly by requiring that each rule (occurrence) shares the special form of the rules in Example 4.5, in which all components of rules are subgraphs of the graph $b \to d \to a \to c$; this essentially amounts to requiring that the components of rules must be subobjects of some (type) object. Indeed, the counterpart of labeled partial orders in the setting of double pushout rewriting will be (a certain class of) *oriented transformation systems*.

☼ Definition 4.6 (Oriented transformation system) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category and let $T \in \mathbb{C}$ be an object, referred to as *type object*. An *oriented transformation system* (OTS) *from S to E (over T)*, written $Y\colon S \rightsquigarrow E$, is a triple $Y = \langle S \hookleftarrow s \rightarrow T, Q, E \hookleftarrow e \rightarrow T\rangle$ where $s\colon S \hookrightarrow T$ and $e\colon E \hookrightarrow T$ are $\mathscr{M}$-morphisms, which are referred to as *start* and *end* object, respectively (they give rise to $\mathscr{M}$-subobjects $[e]$ and $[s]$, respectively). Finally, Q is a set of $\mathscr{M}$-linear $\mathbb{C}\downarrow T$-rules which induce $\mathrm{Sub}_{\mathscr{M}}(T)$-rules, i.e. for each rule $q = (l \leftarrow \alpha \rightarrow k \leftarrow \beta \rightarrow r) \in Q$, all three components l, k and r are $\mathscr{M}$-morphisms. ☼

These oriented transformation systems are very similar to the subobject transformation systems of [Corradini et al., 2008] equipped with start and end objects.[45] Of particular interest in the context of this thesis will be those OTSs which correspond to (switch-equivalence classes of) derivations. The algorithm for obtaining an OTS from a given derivation that faithfully represents the switch-equivalence class that is represented by the derivation can be described as follows. The algorithm proceeds by "collecting" rule occurrences over a type object which "contains" all (instances of) "resources" that are "used" at some point in the derivation; this is sketched in the following example.

Working in the category of $\mathbb{G}\mathrm{raphs}$, the starting point is a certain derivation $\underset{\sim}{\mathfrak{X}}\colon A \models\!\mathscr{R}\!\Rightarrow B$ in $\mathrm{DPO}(\mathbb{G}\mathrm{raphs}, \mathscr{R})_{\mathscr{M}}$, and the result is the corresponding trace process $\langle\!|\mathfrak{X}|\!\rangle$ in $\mathbb{G}\mathrm{raphs}\downarrow T$ where the type graph T is *minimal* as made precise in Definition 4.16. Until properly defined, 'trace process' will be used informally and then refers to some oriented transformation system (which contains enough information to represent a derivation up to switch-equivalence).

❦ Example 4.7 (Trace process of a derivation) In the category of $\mathbb{G}\mathrm{raphs}$, let $q_1 =$ ↺ⓤ▷ⓥ $\leftarrow\!\!\prec$ ⓤ ⓥ $\rightarrowtail$ ⓤ▷ⓥ↻ and $q_2 =$ ⓤ▷ⓥ $\leftarrow\!\!\prec$ ⓤ ⓥ $\rightarrowtail$ ⓤ◁ⓥ be two rules, which one might want to think of as modeling the dispatching of a message over a channel, and the reversing of a channel, respectively. Next take the graph ↺ⓐ◁ⓑ◁ⓒ as the domain of the derivation. The rules roughly mimic the scheme of walkie-talkie communication, as loops can be "transmitted" along edges while each edge can be reversed to switch the roles of "sender" and "receiver". As an example, consider the following sequence of derivation.



[45]This allows to define a category of oriented transformation systems as made explicit in Definition C.1.

It starts from S and leads to E as displayed. The corresponding trace process is illustrated in Figure 47. It contains one *rule occurrence* for each rule application

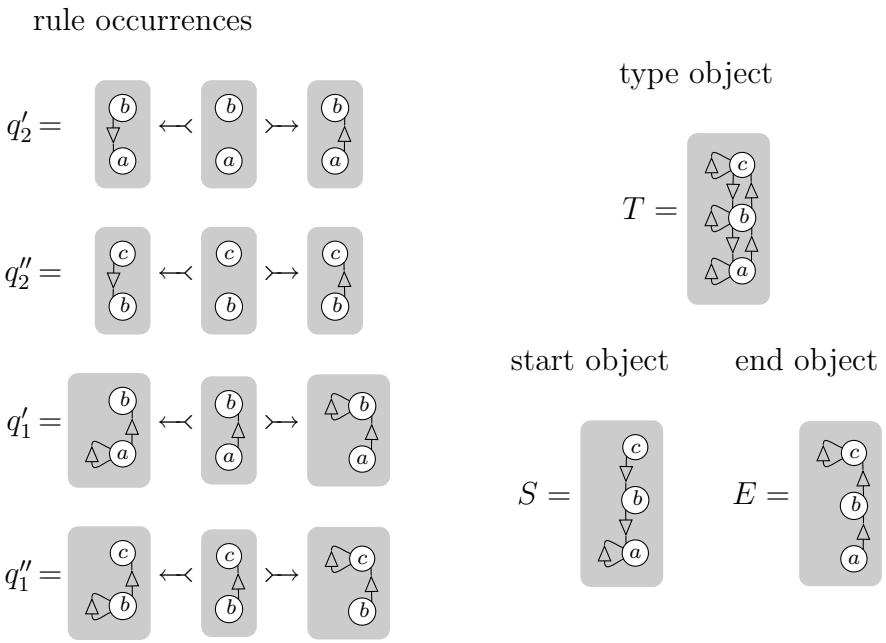


Figure 47: Illustration of the trace process of Example 4.7

of the sequence, namely q_2', q_2'', q_1', and q_1''; the latter are all "included" in the type object T, which contains all nodes and edges that are either present in the start object S or have been created by one of the four applications of rules. Finally, the end object E is fixed. Note, that indeed S and E, and also all rule components are typed over T via "graphical" embeddings (in other words, all these define objects in the slice category $\mathbb{G}\mathtt{raphs}\downarrow T$ that are monic).[46]

One crucial point of the resulting oriented transformation system is the fact that every derivation of length four (applying each of q_1', q_1'', q_2' and q_2'' exactly once) induces a derivation which is switch-equivalent to the original one. For example, the rule q_2'' could also be applied one step earlier or later. The partial order of causality between rule occurrences is completely specified by $q_2' \preceq q_1'$, $q_2'' \preceq q_1''$ and $q_1' \preceq q_1''$.

[46]In this example, to enhance readability, the so-called *typings* of the rule occurrences are given implicitly (the details of the process construction are given in Definition 4.8); e.g. the graphical representation $\textcircled{a}\text{-}\!\lhd\!\text{-}\textcircled{b}$ of the left hand side of q_2' stands for the graph morphism into the type object T with source graph $\textcircled{v}\text{-}\!\lhd\!\text{-}\textcircled{u}$ which maps v to a and u to b.

These dependencies arise "automatically" by construction of T as the "union" of all nodes and edges that occur in the derivation and the "embedding" of the rules into T. Indeed, all "reschedulings" are compatible with the above partial order; the reason is the "additional" requirement that each rule (occurrence) must be applied in the category of T-typed graphs, i.e. in the slice category $\mathbb{Graphs}\downarrow T$.

Finally, to precisely obtain a derivation in the category DPO($\mathbb{Graphs}$) from such an *execution*[47] of the trace process in Figure 47, which is performed in the slice category $\mathbb{Graphs}\downarrow T$, it is necessary to apply the forgetful functor $|\_|\colon \mathbb{Graphs}\downarrow T \to \mathbb{Graphs}$ to all objects that are involved in the derivation where the forgetful functor $|\_|$ "projects" each typed graph $(G{-}g{\to}T) \in \mathbb{Graphs}{\downarrow}T$ to $G \in \mathbb{Graphs}$, i.e. $|G{-}g{\to}T| = G$. The resulting derivation is then switch-equivalent to the original derivation (up to isomorphism) as made precise in Theorem 4.14 below.

Trace processes of derivations will turn out to be nothing else but a particular class of oriented transformation systems which can be characterized statically as described in Section 4.3. Note how the embeddings of the rule components into the type object T implicitly give the causal relations between the rules; in other words, using the slice category $\mathbb{C}\downarrow T$ as ambient category implicitly specifies the causal dependencies.

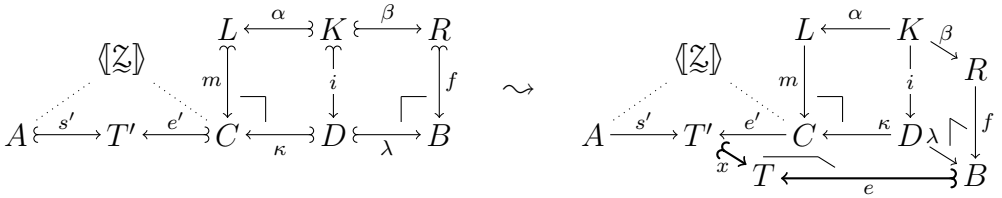
In an abstract weakly $\mathscr{M}$-adhesive category, the type object T, i.e. the object which "contains" all "resources" that are "used" in a derivation, can be constructed by taking successive pushouts as described next in Definition 4.8. The construction of trace process of a DPO-derivation in $\mathbb{C}$ will yield a particular oriented transformation system with a set of rules in the slice category $\mathbb{C}\downarrow T$. During the construction of the trace process, each rule application in the derivation gives rise to a new rule occurrence in the trace process. Since "new resources" are "added" to the type object successively during the construction, all rule occurrences that have already been present must be adjusted to the larger type object after each addition of a new rule occurrence. In fact, Example 4.7 is actually a trace process in the sense of the following definition.

☼ DEFINITION 4.8 (Trace process of a derivation) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $\mathscr{R}$ be a set of $\mathscr{M}$-linear rules, and let $\mathfrak{X}\colon A \models\!\!\overset{\mathscr{R}}{\Rightarrow} B$ be a derivation in DPO$(\mathbb{C},\mathscr{R})_{\mathscr{M}}$. Then a *trace process of* $\mathfrak{X}$, denoted by $\langle\!|\mathfrak{X}|\!\rangle$, can be obtained by the following recursive procedure.

In case that $\mathfrak{X}$ is the empty derivation $\mathfrak{I}_A\colon A \models\!\!\overset{\mathscr{R}}{\Rightarrow} A$ in DPO$(\mathbb{C},\mathscr{R})_{\mathscr{M}}$, then

[47] Executions will be defined formally in Definition 4.13.

$\langle\![\mathfrak{X}]\!\rangle = \langle A -\mathrm{id}\to \cdot, \varnothing, A -\mathrm{id}\to \cdot\rangle$ in $\mathbb{C}\!\downarrow\! A$ is a trace process; otherwise, $\mathfrak{X}\colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$ is a non-empty sequence of direct derivation diagrams, and it factors uniquely as $\mathfrak{Z} \fatsemi \mathfrak{Y}$ where $\mathfrak{Y}\colon C \models\!\!\!\Rightarrow_{q} B$ is a direct derivation diagram for some rule $q = (L \leftarrow\!\alpha\!\rightarrowtail K \rightarrowtail\!\beta\!\leftarrow R) \in \mathscr{R}$. Suppose that $\langle\![\mathfrak{Z}]\!\rangle = \langle A -s'\to \cdot, Q', C -e'\to \cdot\rangle$ in $\mathbb{C}\!\downarrow\! T'$ is a trace process of $\mathfrak{Z}$; now, construct a new type object T as follows.



In other words let $T' -x\to T \leftarrow e- B$ be a pushout of $T' \leftarrow e'\circ\kappa - D -\lambda\to B$. Then $\langle\![\mathfrak{X}]\!\rangle := \big\langle (x \circ s'),\, (\{q_T\} \cup \Sigma_x(Q')),\, e\big\rangle$ in $\mathbb{C}\!\downarrow\! T$ is a trace process of $\mathfrak{X}$ where $q_T = (x \circ e' \circ m) \leftarrow\!\alpha\!\rightarrowtail (e \circ \lambda \circ i) \rightarrowtail\!\beta\!\leftarrow (e \circ f)$ is the new rule occurrence of q and the functor $\Sigma_x\colon \mathbb{C}\!\downarrow\! T' \to \mathbb{C}\!\downarrow\! T$ "adjusts" the set Q' of rules in $\mathbb{C}\!\downarrow\! T'$ by "post-composing" each rule in Q' with the morphism $x\colon T' \to T$, i.e.

$$\Sigma_x(Q') := \Big\{ (x \circ l) \xleftarrow{\alpha} (x \circ k) \xrightarrow{\beta} (x \circ r) \;\Big|\; \Big(l \xleftarrow{\alpha} k \xrightarrow{\beta} r\Big) \in Q' \Big\}.$$

It is immediate that, by construction, each trace process of a derivation $\mathfrak{X}\colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$ in $\textsc{dpo}(\mathbb{C}, \mathscr{R})_{\mathscr{M}}$ is an oriented transformation system of the form $\langle\![\mathfrak{X}]\!\rangle = \langle s\colon A \hookleftarrow T, Q, e\colon B \hookleftarrow T\rangle$ for some set of rules in $\mathbb{C}\!\downarrow\! T$ (for some object $T \in \mathbb{C}$); moreover all morphisms mentioned in the definition are actually $\mathscr{M}$-morphisms. Trace processes are closely related to the concatenable processes of [Baldan, 2000] and the latter terminology will be reused for trace processes up to isomorphism (see Definition 4.16). Here, the focus is on the analogy with Mazurkiewicz traces which presents trace processes as the counterpart of labeled partial orders.

Lemma 4.9 ($\mathscr{M}$-typing of trace processes) Let $\mathfrak{X}\colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$ be a derivation in $\textsc{dpo}(\mathbb{C}, \mathscr{R})_{\mathscr{M}}$ and let $\langle\![\mathfrak{X}]\!\rangle = \langle s\colon A \to T, Q, e\colon B \to T\rangle$ be a trace process of the derivation $\mathfrak{X}$.

Then $\langle s\colon A \to T, Q, e\colon B \to T\rangle$ is an oriented transformation system, i.e. the start and end object are $\mathscr{M}$-morphisms ($s, e \in \mathscr{M}$), and each rule $(l \leftarrow\!\alpha\!- k -\!\beta\!\rightarrow r) \in Q$ is a $\mathbb{C}\downarrow T$-rule with $\mathscr{M}$-morphisms as components ($l, k, r \in \mathscr{M}$).

Moreover, the construction of trace processes is essentially functional and can actually be extended to a functor from the category of dpo-derivations to

a category of oriented $\mathscr{M}$-processes (up to isomorphism), which is very similar to the category of concatenable processes of [Baldan, 2000]. In analogy with the fact that pomsets are actually defined as isomorphism classes of labeled partial orders, it is convenient to identity two trace processes that only differ in the choices of pushouts during the construction. The intended meaning of sameness is isomorphism of oriented transformation systems.

☼ Definition 4.10 (Isomorphism of OTSs) Let $\mathbb{C}$ be any weakly $\mathscr{M}$-adhesive category. Then an *isomorphism* between an oriented transformation system $Y_1 = \langle S \hookleftarrow s_1 \rightarrow T_1, Q_1, E \hookleftarrow e_1 \rightarrow T_1 \rangle$ over T_1 and another oriented transformation systems $Y_2 = \langle S \hookleftarrow s_2 \rightarrow T_2, Q_2, E \hookleftarrow e_2 \rightarrow T_2 \rangle$ over T_2 is an $\mathbb{C}$-isomorphism $i \colon T_1 \xrightarrow{-\cong} T_2$ that respects their structure, i.e. such that the equations $s_2 = i \circ s_1$, $e_2 = i \circ e_1$, and $Q_2 = \Sigma_i(Q_1)$ hold where

$$\Sigma_i(Q_1) = \left\{ (i \circ l) \overset{\alpha}{\leftarrow} (i \circ k) \overset{\beta}{\rightarrow} (i \circ r) \;\middle|\; \left(l \overset{\alpha}{\leftarrow} k \overset{\beta}{\rightarrow} r \right) \in Q_1 \right\}.$$

Two typed oriented transformation systems Y_1, Y_2 are *isomorphic*, written $Y_1 \cong Y_2$, if there exists an isomorphism between them. ☼

Whereas (the labeled partial orders of) pomsets have an explicit representation of the dependency between letter occurrences, the dependencies between rule occurrences of a trace process are recorded implicitly. However the embeddings of the components of the rules into the type object provide information that exactly captures the inherent dependencies of the rule occurrences. In formal terms, the following theorem can be obtained (see Appendix C for a proof).

◌ Theorem 4.11 (Switch-equivalence via trace processes) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category let $\mathscr{R}$ be a set of linear rules. Further let $\underset{\sim}{\mathfrak{X}}, \underset{\sim}{\mathfrak{Z}} \colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$ be two derivations in $\mathrm{DPO}(\mathbb{C}, \mathscr{R})_{\mathscr{M}}$, and let $\langle\![\underset{\sim}{\mathfrak{X}}]\!\rangle$ and $\langle\![\underset{\sim}{\mathfrak{Z}}]\!\rangle$ be two trace processes of $\underset{\sim}{\mathfrak{X}}$ and $\underset{\sim}{\mathfrak{Z}}$, respectively.

Then the two trace processes $\langle\![\underset{\sim}{\mathfrak{X}}]\!\rangle$ and $\langle\![\underset{\sim}{\mathfrak{Z}}]\!\rangle$ are isomorphic if and only if the derivations $\underset{\sim}{\mathfrak{X}}$ and $\underset{\sim}{\mathfrak{Z}}$ are switch-equivalent to each other (up to isomorphism).

The central task consists in proving the special case in which $\underset{\sim}{\mathfrak{X}}$ and $\underset{\sim}{\mathfrak{Z}}$ form a switching couple. In other words, the central fact at the heart of the preceding theorem is the following proposition, which gives a "more conceptual" characterization of switching couples.

◌ Proposition 4.12 (Switching couples via trace processes) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category let q', q be a pair of $\mathscr{M}$-linear rules. Further let $\mathfrak{X}_1 \colon A \models\!\!\!\Rightarrow_{q'} C$ with $\mathfrak{Y}_2 \colon C \models\!\!\!\Rightarrow_{q} B$, and $\mathfrak{Y}_1 \colon A \models\!\!\!\Rightarrow_{q} C'$ with $\mathfrak{X}_2 \colon C' \models\!\!\!\Rightarrow_{q'} B$ be two composable pairs of sequential-independent direct derivation diagrams

yielding derivations $\mathcal{X}_1 \fatsemi \mathcal{Y}_2, \mathcal{Y}_1 \fatsemi \mathcal{X}_2 \colon A \Longrightarrow B$ in $\mathrm{DPO}(\mathbb{C})_{\mathscr{M}}$. Finally, let $\langle\!\langle \mathcal{X}_1 \fatsemi \mathcal{Y}_2 \rangle\!\rangle$ and $\langle\!\langle \mathcal{Y}_1 \fatsemi \mathcal{X}_2 \rangle\!\rangle$ be trace processes of $\mathcal{X}_1 \fatsemi \mathcal{Y}_2$ and $\mathcal{Y}_1 \fatsemi \mathcal{X}_2$, respectively.

Then $\langle \mathcal{X}_1\mathcal{Y}_2, \mathcal{Y}_1\mathcal{X}_2 \rangle$ is a switching couple if and only if their trace processes are isomorphic, i.e. if and only if $\langle\!\langle \mathcal{X}_1 \fatsemi \mathcal{Y}_2 \rangle\!\rangle \cong \langle\!\langle \mathcal{Y}_1 \fatsemi \mathcal{X}_2 \rangle\!\rangle$.

Though Theorem 4.11 shows that trace processes are a suitable means to represent switch-equivalence classes of derivations, one might wonder whether there is a constructive way to go back and forth between (switch-equivalence classes of) derivations and processes.

4.2.1 From derivations to trace processes and back

Before addressing the issue of a static characterization of those oriented transformation systems that arise as trace processes of derivations in Section 4.3, executions of a trace processes are described. Indeed, these executions will be shown to provide a means to recover representatives of trace processes. The main result is the analogue of the rather obvious fact that each pomset $[\langle V, \preceq, \Sigma, \lambda \rangle]$ that arises as the result of the *string-to*-LPO algorithm, can be linearized. This means that each of the underlying partial orders $\langle V, \preceq, \Sigma, \lambda \rangle$ has a linearization $v_1 \preceq \cdots \preceq v_n$ of length $n = |V|$ such that $V = \{v_1, \ldots, v_n\}$. Moreover, given any such linearization, application of the *string-to*-LPO algorithm to the word $\lambda(v_1) \cdots \lambda(v_n) \in \Sigma^*$ yields an isomorphic labeled partial order $\langle V', \preceq', \Sigma, \lambda' \rangle \cong \langle V, \preceq, \Sigma, \lambda \rangle$, and hence the pomset $[\langle V', \preceq', \Sigma, \lambda' \rangle]$ is exactly the same as the original one $[\langle V, \preceq, \Sigma, \lambda \rangle]$. In the realm of double pushout rewriting, the role of the *linearization* of a labeled partial order is played by the *execution* of an oriented transformation system.

☼ Definition 4.13 (OTS executions) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, and let $Y = \langle S \hookleftarrow s \rightarrow T, Q, E \hookleftarrow e \rightarrow T \rangle$ be an oriented transformation system.

A *(typed) execution of* Y is defined as a derivation $\mathfrak{X} \colon s \Longrightarrow_Q e$ in the derivation category $\mathrm{DPO}(\mathbb{C}\downarrow T, Q)_{\mathscr{M}}$ in which each rule is used exactly once, which means that for each rule $q \in Q$, the derivation $\mathfrak{X}$ uniquely factorizes as $\mathfrak{X}' \fatsemi \mathcal{Z} \fatsemi \mathfrak{X}'' = \mathfrak{X}$ such that $\mathcal{Z} \colon a \Longrightarrow_q b$ is a direct derivation diagram (for some pair of objects $(A \rightarrow a \rightarrow T), (B \rightarrow b \rightarrow T) \in \mathbb{C}\downarrow T$).[48] ☼

Note that the notion of execution and OTS might appear overly general: there are (non-trivial) OTSs without any executions, and – in contrast to the subobject transformation systems of [Corradini et al., 2008] – not all objects that are reached in an execution have to be monomorphisms. However, as a

[48] That it is common to use the letters $a, b, c, \ldots$ not only to range over letters of an alphabet Σ but also to denote objects of a (slice) category $\mathbb{C}\downarrow T$ is merely a coincidence.

consequence of results that will be established later, one could restrict attention to executable and "safe" oriented transformation systems to avoid these issues. Independent of these issues, a crucial point of the definition is that executions are performed in the slice category over the type object. The corresponding untyped derivation is obtained by forgetting the type information, i.e. by application of the forgetful functor.

The concept of executions of a trace processes allows to give the statement that all executions of a trace process of some derivation are switch equivalent to the original one (up to isomorphism) a precise meaning. This will be formulated in the first part of the next theorem. Moreover, also the converse holds, i.e. given a derivation $\mathfrak{X}$, any switch equivalent derivation $\mathfrak{X}' \approx_{\mathrm{sw}} \mathfrak{X}$ corresponds to some execution of the trace process $\langle\![\mathfrak{X}]\!\rangle$.

THEOREM 4.14 (Execution as inverse to trace process construction) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $\mathscr{R}$ be a set of $\mathscr{M}$-linear rules, let $\mathfrak{X}\colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$ be a derivation in $\mathrm{DPO}(\mathbb{C},\mathscr{R})_{\mathscr{M}}$, and let $\langle\![\mathfrak{X}]\!\rangle = \langle a, Q, b\rangle$ be a trace process of the derivation $\mathfrak{X}$. Then the following hold.

1. Any execution $\mathfrak{Z}\colon a \models\!\!\!\Rightarrow_{Q} b$ of $\langle\![\mathfrak{X}]\!\rangle$ induces a derivation that is (essentially) switch-equivalent to $\mathfrak{X}$, or more precisely, $\mathfrak{X} \approx_{\mathrm{sw}} \mathfrak{Z}' \cong |\_| \circ \mathfrak{Z}$ holds for some derivation $\mathfrak{Z}'\colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$.

2. Conversely, all derivations that are (essentially) switch equivalent to $\mathfrak{X}$ can be obtained by execution of $\langle\![\mathfrak{X}]\!\rangle$, or more precisely if $\mathfrak{Z}' \approx_{\mathrm{sw}} \mathfrak{X}$ holds for some derivation $\mathfrak{Z}'\colon A \models\!\!\!\Rightarrow_{\mathscr{R}} B$, then there exists an execution $\mathfrak{Z}\colon a \models\!\!\!\Rightarrow_{Q} b$ of $\langle\![\mathfrak{X}]\!\rangle$ such that $\mathfrak{Z}' \cong |\_| \circ \mathfrak{Z}$.

4.3 A static characterization of trace processes

A declarative definition of *oriented $\mathscr{M}$-processes*[49] would describe them as those oriented transformation systems that arise by applying the trace process construction of Definition 4.8 to some derivation. However, directly checking this definition for a given OTS would amount to finding an execution of it and verifying whether the OTS is isomorphic to a trace process of this execution;[50] as a direct consequence, this approach is *not static* because it

[49] These will turn out to be essentially the same as the finite deterministic processes of [Baldan et al., 2006a], which date back to work on Petri nets [Goltz and Reisig, 1983].

[50] As a simple example of an OTS which is not isomorphic to the trace process of its execution consider a Petri net with one place which is both the pre- and the post-set of a single transition; the start and the end marking are a single token. The trace process of the

involves the *execution* of an OTS. Now the question is, whether the *static* characterizations of processes known from Petri nets [Goltz and Reisig, 1983] and graph transformation systems [Montanari et al., 1996] can be lifted to the abstract level of OTSs in weakly $\mathscr{M}$-adhesive categories.

As a first sketch of the general form of the result to be established, consider once more the case of pomsets: also they could be described as those structures that arise by applying Algorithm 1 to a string over some dependency alphabet $\langle \Sigma, D \rangle$; also in this case, directly checking this property would involve the "execution" of a given labeled partial order $\langle V, \preceq, \Sigma, \lambda \rangle$ (the "execution" of $\langle V, \preceq, \Sigma, \lambda \rangle$ is just a linearization of the partial order $\preceq$), followed by the application of Algorithm 1 and an isomorphism check. Alternatively, pomsets can be characterized as the finite *dependence graphs* – a direct consequence of of [Diekert and Métivier, 1997, Proposition 6.2].

With this example in mind, the goal is to find an analogon to the notion of dependence graph in the realm of DPO-rewriting. Its name, viz. *oriented $\mathscr{M}$-process*, is based on the analogies with the existing theory, e.g. the Petri net processes [Goltz and Reisig, 1983] or graph processes [Montanari et al., 1996]. Here, the emphasis lies on the possibility of a '*static*' description, i.e. a characterization of processes that does not involve the notion of execution; as a fact, this static approach allows for efficient analysis techniques for the concrete cases of Petri nets [Baldan et al., 2008e] and graphs [Baldan et al., 2008c].

The central common feature of dependence graphs and oriented $\mathscr{M}$-processes is the absence of cycles in the dependency relation, where the dependence relation in oriented $\mathscr{M}$-processes is determined by *causality* and *(asymmetric) conflict* between rule occurrences. These can be defined in any oriented transformation system.[51]

☼ DEFINITION 4.15 ((Co-)causality and (co-)asymmetric conflict) Let $\langle s, Q, e \rangle$ be an oriented transformation system. A pair of rules $q = l_q \leftarrow\!\alpha_q\!\rightarrow k_q \leftarrow\!\beta_q\!\rightarrow r_q$ and $q' = l_{q'} \leftarrow\!\alpha_{q'}\!\rightarrow k_{q'} \leftarrow\!\beta_{q'}\!\rightarrow r_{q'}$ in Q, i.e. any pair $q, q' \in Q$, may be related in any of the following ways.

$\prec$: *q directly causes q'*,	written $q \prec q'$,	if $r_q \sqcap l_{q'} \not\sqsubseteq k_q$
$\ll$: *q can be disabled by q'*,	written $q \ll q'$,	if $l_q \sqcap l_{q'} \not\sqsubseteq k_{q'}$

unique execution is another OTS with one transition but two copies of the "original" place: one copy is the pre-set and the other the post-set.

[51] More precisely, these notions are applicable to any set of rules which are *mono-typed*. This means, that (co-)causality and asymmetric conflict can be defined w.r.t. any rule set Q in some slice category $\mathbb{C}\!\downarrow\!T$ such that all components of a rule $(l \hookleftarrow k \hookrightarrow r) \in Q$ are monomorphisms and thus give rise to subobjects $[l], [k]$, and $[r]$ in $\mathrm{Sub}(T)$.

Further, the *asymmetric conflict* relation is $\nearrow := \prec^+ \cup (\ll \setminus \mathrm{id}_Q)$ where $\prec^+$ is the transitive closure of $\prec$, and $\ll \setminus \mathrm{id}_Q$ is the irreflexive restriction of $\ll$. Further, applying these relations to the *opposite* rules yields the corresponding *co*-relations: whenever $\propto$ is any of the above binary relations $\prec, \ll$ or $\nearrow$, then, by definition, $q_1 \propto_\circ q_2$ if $q_2^\circ \propto q_1^\circ$ for the opposite rules q_2° and q_1° with swapped left and right hand sides.

Similarly, given a subobject $[a] \in \mathrm{Sub}(T)$, its (direct) *causes* and *co-causes* are $\llcorner a \lrcorner = \{q \in Q \mid r_q \sqcap a \not\sqsubseteq k_q\}$ and $\ulcorner a \urcorner = \{q \in Q \mid l_q \sqcap a \not\sqsubseteq k_q\}$, respectively; further, the *(complete) causes* and *co-causes* of the subobject $[a]$ are $\lfloor a \rfloor = \{q \in Q \mid \exists q' \in \llcorner a \lrcorner .\, q \prec^* q'\}$ and $\lceil a \rceil = \{q \in Q \mid \exists q' \in \ulcorner a \urcorner .\, q \prec_\circ^* q'\}$, respectively. Finally, for any subset $Q' \subseteq Q$, its *downward* and *upward closure* is $\lfloor Q' \rfloor = \{q \in Q \mid \exists q' \in Q' .\, q \prec^* q'\}$ and $\lceil Q' \rceil = \{q \in Q \mid \exists q' \in Q' .\, q \prec_\circ^* q'\}$, respectively. ☼

Having these notions concerning causality at hand, a sufficient and complete list of characteristic properties of trace processes is summarized in the definition of *concatenable process* (see Definition 4.16). Apart from the acyclicity of asymmetric conflict $\nearrow$ in both directions, further additional properties make formal in what sense the rules fully determine causality in the process.

As will be stated in Theorem 4.17, in any adhesive category, these three notions, namely *acyclicity* of asymmetric conflict with *soundness* and *completeness* of causality, provide the vocabulary for a static characterization of all OTS that arise from the trace process construction. In other words, there is an exact correspondence between OTSs satisfying these properties and (isomorphism classes) of trace processes. But even for the more general case of weakly $\mathscr{M}$-adhesive categories, which may fail to have effective unions, a single extra condition suffices, which ensures that all relevant joins can be "constructed" by taking pushouts over intersections.

☼ Definition 4.16 (Concatenable DPO-process) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, and let $A, B \in \mathbb{C}$.

A *concatenable process from A to B* is an isomorphism class $[\langle s, Q, e\rangle]$ of an oriented transformation system $\langle s, Q, e\rangle \colon A \rightsquigarrow B$ over $T \in \mathbb{C}$ which satisfies the following conditions.

(CO-)ACYCLIC: The set of rules Q is finite and there are no cycles in the relation of asymmetric conflict $\nearrow$ and co-asymmetric conflict $\nearrow_\circ$.

(CO-)CAUSALLY SOUND: The start object has no causes and the end object has no co-causes, i.e. $\lfloor s \rfloor = \varnothing = \lceil e \rceil$.

(CO-)CAUSALLY COMPLETE: Each subobject is covered by its causes and its co-causes, i.e. for any $\mathscr{M}$-subobject $[w] \in \mathrm{Sub}_{\mathscr{M}}(T)$, the two inclusions $[w] \sqsubseteq s \sqcup \bigsqcup_{q \in \lfloor w \rfloor} r_q$ and $[w] \sqsubseteq e \sqcup \bigsqcup_{q \in \lceil w \rceil} l_q$ hold.

If $\mathbb{C}$ does *not* have $\mathscr{M}$-effective unions, the following must be required explicitly:

(CO-)DISSECTABLE: The two joins $s \sqcup \bigsqcup_{q \in \lfloor Q' \rfloor} r_q$ and $e \sqcup \bigsqcup_{q \in \lceil Q' \rceil} l_q$ exist in $\mathrm{Sub}(T)$ for any set $Q' \sqsubseteq Q$ and these joins are actually $\mathscr{M}$-subobjects, i.e. elements of $\mathrm{Sub}_{\mathscr{M}}(T)$.

Whereas the first three properties only concern causality, the last one ensures that the rule components and the start and end object can be used as building blocks and thus the whole process can be dissected into smaller ones. Now the exact formulation of the static characterization of trace processes reads as follows.

THEOREM 4.17 (Static description of concatenable processes) Given an OTS $[\langle s\colon A \hookleftarrow T, Q, e\colon B \hookleftarrow T\rangle]$ without idling rules, i.e. such that for all rules $q = (l \leftarrow\!\alpha\!- k -\!\beta\!\rightarrow r) \in Q$ not both of α and β are isomorphisms, there exists a derivation $\mathcal{X}\colon A \rightleftharpoons^{|Q|} B$ in $\mathbb{C}$ such that $\langle s, Q, e\rangle$ is a trace processes of $\mathcal{X}$ if and only if $[\langle s, Q, e\rangle]$ is a concatenable process.

REMARK 4.18 (Relation to concepts in the literature) The concepts and results of the present section are essentially generalizations of previous work on concatenable processes for Petri nets and graph grammars (see [Baldan et al., 1998a] and [Degano et al., 1996], respectively). Some subtle differences concern the role of isomorphisms of derivations and the definition of the category of traces and concatenable processes. A precise treatment of isomorphisms of derivations is given in the appendix (see Definition C.12). As for the category of concatenable processes, in [Baldan et al., 1998a] isomorphism classes of graphs are used as objects (and not just graphs). The resulting additional complexity is avoided in the present thesis to obtain a more accessible presentation of the central concepts.

This concludes the discussion of processes as the representation of DPO-derivations. The definitions and the theory of this section can be adapted in a straightforward manner to single pushout rewriting. The main difference would concern the reversibility of processes. Whereas each DPO-derivation is reversible, the same does not hold for SPO-derivations due to the more radical deletion mechanism of the latter. As a result also SPO-processes are not

reversible. However, the SPO-rewriting features prominently in the unfolding of adhesive rewriting systems.

5 *Concurrent evolutions of systems as unfoldings*

Concurrency is a phenomenon that does not only manifests itself in isolated (finite) computations/runs of (distributed) systems but it remains relevant for the analysis of *all* possible evolutions of a given system which depart from a fixed start state. To give an exhaustive description of the totality of all possible changes in rule or event based concurrent systems – take Petri nets or graph transformation systems as paradigmatic examples – the *unfolding* of systems has become an established concept. Such an unfolding of a system explicitly contains all possible applications/occurrences of rules/events and additional implicit information concerning their mutual dependencies.

Indeed, unfoldings can be used to provide *truly concurrent* semantics for distributed systems; the fundamental idea is essentially the same as the one of trace processes, namely to consider computations/runs up to concurrency. In particular, no spurious interleavings of concurrent events are introduced, and hence "all concurrency" that is present in a given system is still present in its unfolding. One qualitative difference between processes and unfoldings is determinism: while the former are essentially deterministic, the latter have to account for choice points and bifurcations, which typically are present in models of systems with several different consumers that compete for limited resources.

The preservation of concurrency is achieved by only recording the necessary causal dependencies of all the events that may possibly occur in some system run; the resulting causality relation is acyclic and actually a partial order. Choice points are represented implicitly by a conflict relation between rule occurrences, which can be derived from causality. As a "side effect", the preservation of concurrency and the implicit modeling of choice points make unfoldings "efficient" or "compact" representations of systems. And in fact, this is also one of the reasons, why unfoldings have become a standard concept in the area of partial order verification techniques (see e.g. [Esparza and Heljanko, 2008]).

This "compactness" of unfoldings provides a (partial) explanation of their fruitfulness; it may be illustrated by a comparison with fully explicit transition systems of rule or event based system specifications. As a simple, concrete example, take the transition system of the Petri net that consists of n "parallel" copies of a single enabled transition ◉→□→○;[52] while this net (which will turn out to be identical to its net unfolding) has size $3n$, the explicit transition

[52] Alternatively, one could take a net consisting of one transition and one place which is the preset of the transition and contains n tokens.

system has 2^n states; this means that the state space is exponentially larger than the unfolded net. This significant difference in size of transition systems and unfoldings is due to the use of the partial order of causality (which is empty in this case); using causality in this way leads to the "automatic" identification of different interleavings of a given computation (up to concurrency), and indeed, each actual system run will turn out to correspond to a sub-process of the whole unfolding. This example illustrates how the concept of unfolding allows to avoid the state explosion problem whenever a system consists of many relatively small independent subsystems, and thus provides a significantly more compact representation.

Summarizing the previous paragraph, though unfoldings explicitly contain all possible events that may eventually happen (and hence are possibly infinite structures in general), under certain finiteness conditions or when restricting to suitable finite sub-systems of a given system, they allow for a relatively concise but at the same time sufficiently concrete representation of systems.

The statement that unfoldings are *sufficiently concrete* is meant to refer to the the second crucial property of unfoldings from a practitioners point of view, namely that they allow for practicable model checking algorithms (while avoiding the state explosion problem at the same time). In other words, the unfolding of systems is the basic structure "behind" efficient algorithms for the verification of safety properties of rule based systems such as Petri nets. Hence, to complete the short explanation of why unfoldings have become a corner stone of verification techniques, one might argue that they provide a suitable compromise between compactness and explicitness for the analysis of systems.

A self-contained, gentle introduction to the algorithmic ideas for model checking using unfoldings as a *partial order approach to model checking* can be found in the recent book [Esparza and Heljanko, 2008]. In contrast, the present thesis focuses on unfoldings as a means to provide truly concurrent semantics of systems, and the main contribution is the theoretical foundation for existing [König and Kozioura, 2006a, König and Kozioura, 2006b] and also future work on unfolding based verification techniques for dynamic systems with structured states.

More background concerning these theoretical considerations are presented in Section 6. In contrast, the present section is an attempt to approach the topic from a practitioners angle. From this perspective, the main benefit of the abstract setting of adhesive rewriting systems is that it covers rewriting for a wide range of concrete examples of categories of graph-like structures

that occur in applications.

The present thesis provides a general, uniform unfolding construction which can be applied to all these "real world" structures. More importantly, it provides a proof of the "soundness" of the unfolding semantics and dispenses with the need to reprove it on the concrete level for each single variant of "graph-like" structures – a task that is already non-trivial for the case of graphs [Baldan, 2000]. However, the main difficulty is the solution of a theoretical problem, namely the generalization of the theory of unfoldings in the style of the research initiated by Glynn Winskel [Nielsen et al., 1981], which establishes the unfolding as a so-called coreflective functorial semantics (see e.g. [Bruni et al., 2001]). This topic will be addressed in Section 6.

Overview of the section. Section 5.1 starts with examples of (truncations and finite complete prefixes) of unfoldings based on the influential work of McMillan [McMillan, 1992, McMillan, 1995]. The presentation is informal and only presupposes acquaintance with the very basic notions of Petri nets as reviewed above in Section 1.2. The main goal is to convey the basic ideas of the algorithmic approach to unfoldings of [McMillan, 1995].

The abstract version of this unfolding algorithm of Section 5.1 in any weakly $\mathscr{M}$-adhesive category is given in Section 5.2. Moreover a first "practical" notion of the completeness of unfoldings is discussed: in the case of systems (which satisfy suitable finiteness conditions), every reachable state of the system has a canonical image in the unfolding together with a causal explanation. In other words, the ideas of [McMillan, 1995] are made available for a wide range of systems with "graph-like" structure.

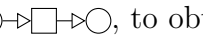
All (intermediate) results of this abstract unfolding algorithm are so-called *occurrence grammars* which are characterized statically in Section 5.3. This class of grammars is the abstract counterpart of occurrence graph grammars and occurrence nets (see e.g. [Baldan, 2000]). A digression on a generalization of McMillan's finite complete prefix is added to emphasize the connection to practical verification techniques such as [Baldan et al., 2008e]. However, the main concept is that of an occurrence grammar, which is a prerequisite for Section 6, which presents the main result of this thesis.

Finally, the unfolding of (countably) infinite state systems is addressed in Section 5.4. The last construction step of the *full unfolding* of a given system amounts to assembling a growing chain of finite prefixes into a single structure. In the concrete cases of graph grammars and Petri nets, this assembly can be understood as the union of all finite prefixes. The categorical abstraction of

these unions are colimits of ω-chains of monomorphisms, which hence have to exist. Moreover, to obtain the main result of this thesis (Theorem 6.6), these colimits are required to satisfy the same properties as pushouts in weakly adhesive categories (see Definition 5.14); categories of this kind are dubbed *weakly $\mathscr{M}\omega$-adhesive*.

5.1 Concrete finite unfoldings and truncations

Though unfoldings are potentially infinite structures in general, their fundamental idea and their role in the context of state space exploration and automatic verification is best illustrated for systems which have a finite transition system without cycles, i.e. for those systems which always terminate. However also finite state systems are instructive examples since the finite complete prefix method [McMillan, 1992, McMillan, 1995] (or generalizations thereof [Baldan et al., 2008e]) gives a fully automatic algorithm which restricts finite state systems to strictly finite subsystems which already "cover" all reachable states.

Now, as a very rough approximation of the fundamental idea of (finite complete prefixes) of unfoldings, the goal is to represent all derivations (up to a fixed length k)[53] within a single structure, such that "common resources" are shared and the derivations "overlap" in a canonical way. In fact, the number of derivations that need to be considered to account for every reachable state might be significantly smaller. For example, in the Petri net that consists of n "parallel" copies of a single enabled transition  , to obtain the full unfolding it is enough to fire each of the n transitions only once "in isolation", i.e. $k = 1$.

The role of unfoldings for state space search have been illustrated in the seminal paper [McMillan, 1995] using the trivial board game illustrated in Figure 48. The board has a number of consecutive fields, and a number of

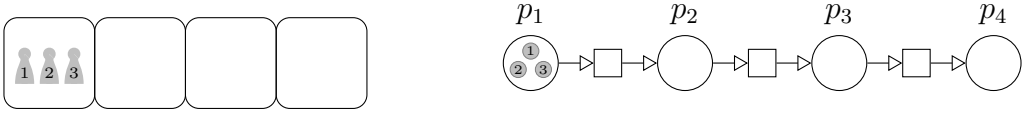


Figure 48: A trivial board game and its Petri net counterpart

[53] More precisely, it is the *causal depth* of all derivations; the latter describes the longest chain of causally dependent events in a given derivation, or more technically the upper bound of the length of chains of the causality relation in the process of the derivation.

pieces in the first place, namely four fields and three pieces in the present example. At each turn of this trivial game, one of the pieces may advance to the next field; the goal of the game is to reach the last field with all pieces. Hence, the Petri net on the right hand side in Figure 48 captures the rules of the game – apart from the winning condition.

It is apparent that no move can block another move, and hence also no move can be blocked by any other move; hence all moves of the game are *persistent* (cf. [McMillan, 1995]); the only restriction is that pieces cannot jump but can move only one step at a time. In particular the order of the moves of a strategy are immaterial and hence the number of different winning strategies for this game considerably decreases: there is only one (up to reordering of moves).

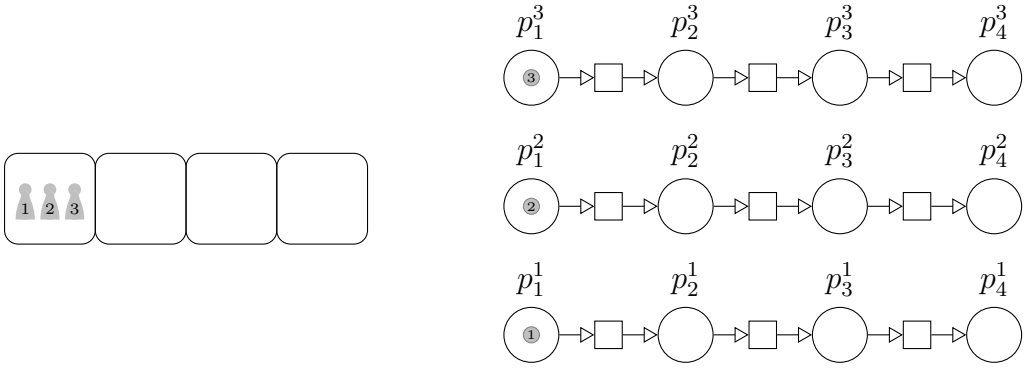
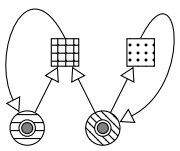


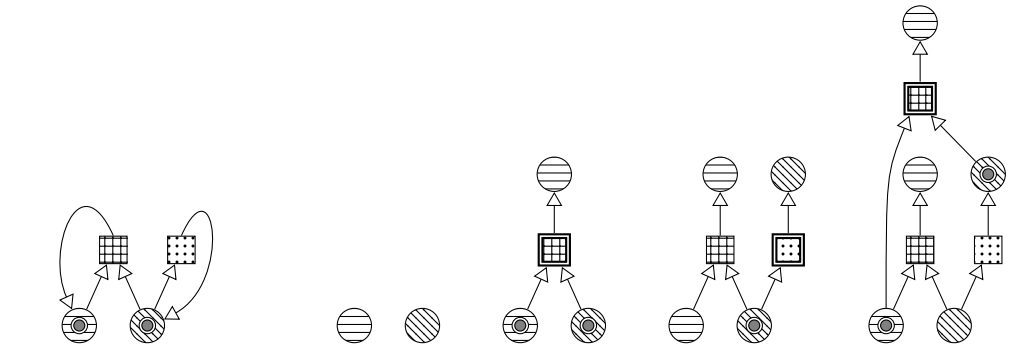
Figure 49: The unfolding of the game

The persistency of moves can actually be derived from the causality relation in the unfolding of the net (see Figure 49). With this causality information at hand, one can perform an "informed search" in the unfolding to determine that it is possible to move all three pieces to the last field; in particular, one can avoid a blind exploration of the exponentially larger state space, which has n^m states where n is the number of fields and m is the number of pieces.



Given a Petri net with a marking, its unfolding is a so-called *occurrence net* which contains a transition for each possible occurrence of any transition in the original net (whence the name). In [McMillan, 1995, Section 2], the unfolding procedure to construct such an unfolding is described as follows. As initialization, make a copy of the places on which tokens reside according to the given marking. For the displayed example net, this just yields one copy of each of the two places ⊖ ⊘ as shown on the left in Figure 50(b). However,

in the board game example, there were three copies of the first place – one for each piece.



(a) Example Petri net with a marking

(b) Three unfolding steps, each one adding one occurrence of a transition

Figure 50: Illustration of the unfolding procedure for Petri nets

After this initialization, which gives an occurrence net with no transitions and only a copy of (the places of) the marking of a given Petri net, the unfolding is constructed by performing the following steps repeatedly.

1. Choose some transition t in the original Petri net.

2. For each place in the *preset* of t, i.e. for each place from which an arrow leads to t, find a copy of it in the occurrence net constructed so far and mark it with a token (if you cannot find enough copies, go back to step 1). As a side condition, to avoid redundancy, the marked places should not have been chosen for t before.

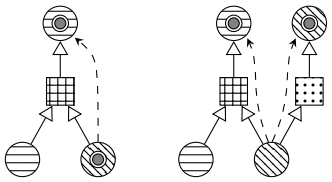
3. For any two places x, y of the marking from step 2, check that

   - there is no path from x to y along the arrows, and also none in the opposite direction from y to x; and that
   - there is no place z in the occurrence net from which you can reach *both* x and y, exiting z towards two different transitions.

   In other words, x and y must be *concurrent* (otherwise got back to step 1).

4. Add a copy of the transition t to the occurrence net and record that it is an occurrence of t (e.g. by naming it t' or the like); moreover connect the new occurrence of t with arcs from the marked places.

5. Add copies of the places in the *post-set* of the transition t to the occurrence net, i.e. adjoin copies of all those places in the original net that have an incoming arc from t; finally add arcs from the new the occurrence of t to the copies of the places in the post-set.



Two examples of markings that are not concurrent are displayed to the right. In the first net, one of the marked places is causally dependent on the other because there is a path from the former to the latter; in the second net, the two marked places are in conflict since there is a third place which causes them via two different successor transitions. Starting from the marking of the original net, i.e. from the marking consisting of one token in each one of the two bottom places, neither of the displayed markings is (coverable by) a reachable marking.

In fact, a marking of an occurrence net is concurrent if and only if it is coverable by a reachable marking. This fundamental fact allows to construct unfoldings statically, i.e. without the need to solve reachability problems. The idea of the line of work presented in [Baldan et al., 2008d] was to transfer the existing analysis and verification methods for Petri Nets to the realm of graph transformation systems. In the same way, they can be generalized to a large class of graph like structures, in particular to functor categories of the form $[\mathbb{C}, \mathbb{Set}]$, i.e. to categories with algebras of multi-sorted signatures with (at most) unary function symbols as objects and algebra homomorphisms as morphisms. Nevertheless, the above five steps of the unfolding procedure can be generalized to the abstract setting of weakly adhesive categories as described next.

5.2 Abstract unfolding of finite systems

Restricting to finite systems with a fixed start state means that there is only a finite set of *reachable states*, namely all those states which can possibly be reached from the start state after the execution of some finite system run. Further restricting to systems which can be faithfully modeled via single or double pushout rewriting in some suitable category $\mathbb{C}$, each event corresponds to the application of some rule $q \in Q$ in a set of rules Q, and the fixed start state is represented by some object $S \in \mathrm{ob}(\mathbb{C})$. In other words, the temporary assumption is that the set of reachable objects is (essentially) finite, i.e. there is a finite set of objects $\{A_1, \ldots, A_n\} \subseteq \mathrm{ob}(\mathbb{C})$ such that the codomain of each

derivation with domain S is (isomorphic to) one of these objects.[54] As an immediate consequence, there is a number $k \in \mathbb{N}$ such that each reachable state can be obtained by "following" a derivation of length at most $k \leq n$.

In the abstract setting of weakly $\mathscr{M}$-adhesive categories, the unfolding procedure takes a pair $\langle Q, S\rangle$ as input where $S \in \mathbb{C}$ is an object (representing a fixed system state) and Q is a set of $\mathscr{M}$-linear rules (representing the "laws" of system evolution). In analogy to Chomsky grammars, such a pair $\langle Q, S\rangle$ is referred to as a *grammar* where S plays the role of the start symbol and set of rules Q is the analogon of the set of productions.

☼ Definition 5.1 (Grammar) Let $\mathbb{D}$ be a category. A *grammar in* $\mathbb{D}$ is a pair $\langle Q, S\rangle$ where Q is a set of $\mathbb{D}$-rules and $S \in \mathbb{D}$ is an object, which is called the *start object*. ☼

However, there is an analogy with Petri nets if the category $\mathbb{D}$ is a slice category of the form $\mathbb{C}\downarrow T$ for some *type object* $T \in \mathbb{C}$. In fact, glossing over some details, a Petri net with a set of places $P \in \mathbb{Set}$ is roughly the same as a grammar in $\mathbb{Set}\downarrow P$ (see also Example 2.4). Hence, if $\langle Q, S \xrightarrow{s} T\rangle$ is a grammar in $\mathbb{C}\downarrow T$, then the object T can be thought of as a generalization of the set of places. Such grammars in slice categories play a central role in the context of this thesis; they are called typed grammars.

☼ Definition 5.2 (Typed grammar) Let $\mathbb{C}$ be a category and $T \in \mathbb{C}$ be an object, referred to as *type object*. A T*-typed grammar in* $\mathbb{C}$ is a pair $\langle Q, s\colon S \to T\rangle$ such that $\langle Q, s\rangle$ is a grammar in the slice category $\mathbb{C}\downarrow T$. ☼

With the definition of typed grammar at hand, the algorithmic description of the unfolding procedure of Petri nets of Section 5.1 can be generalized to all T-typed grammars $G = \langle Q, s\colon S \to T\rangle$ which are *consuming*. This means that rules are not allowed to have isomorphisms on the left, i.e. for all $q = (l_q \xleftarrow{\alpha_q} k_q \xrightarrow{\beta_q} r_q) \in Q$, the morphism α_q is not an isomorphisms. When rules in the original grammar are consuming, the corresponding rule occurrences in the unfolding cannot be executed more that once. This is in complete analogy to the Petri net case. From now on, grammars are assumed to be consuming.

Given a grammar $G = \langle Q, s\colon S \to T\rangle$ in some weakly $\mathscr{M}$-adhesive category where Q is a set of $\mathscr{M}$-linear rules, the initialization phase of the unfolding procedure consists in making a "copy" of the start object s, yielding the

[54] More precisely, such a pair $\langle Q, S\rangle$, ofter referred to as a *grammar*, is considered finite state if for each derivation $\mathcal{X}\colon S \overset{Q}{\Rightarrow} A$ there exists some $A_i \in \{A_1, \ldots, A_n\}$ such that $A \approx A_i$ in $\mathbb{C}$.

domain $S \in \mathbb{C}$ as type object and the empty set of productions. In other words, the starting point of the unfolding procedure is the S-typed grammar $U_0 = \langle \varnothing, \mathrm{id}_S \colon S \to S \rangle$. The intermediate results of the unfolding procedure will be T_i-typed grammars of the form $U_i = \langle Q_i, s_i \colon S \hookleftarrow T_i \rangle$.

Though not mentioned explicitly in the above net examples, there were implicit functions mapping the places (and transitions) of the unfoldings back to the original net. In the board game example of Figure 49, each place p_i^x is mapped to p_i, and in Figure 50, each place in the unfolding is mapped to the unique place in the original net which shares the same pattern. The analogon of these functions will be so-called *folding morphisms* $\lambda_i \colon T_i \to T$ in $\mathbb{C}$, which "fold" the type objects of the unfoldings to the type object of the original grammar. These folding morphisms moreover induce functions which map rule occurrences in Q_i to rules of the original grammar G using the functor $\Sigma_{\lambda_i} \colon \mathbb{C} \downarrow T_i \to \mathbb{C} \downarrow T$ which acts on objects as post-composition with λ.

Now, the counterpart of the five steps in Section 5.1 are the following.

1. Choose a production $q = (l_q \leftarrow \alpha_q \rightarrowtail k_q \rightarrowtail \beta_q \rightarrow r_q) \in Q$ in the original grammar.

2. For the left hand side $l_q \colon L_q \to T$ of the rule q, find an $\mathscr{M}$-morphism $\nu \colon L_q \hookleftarrow T_i$ in the grammar U_i such that $l_q = \lambda_i \circ \nu$ (if you cannot find such a morphism, go back to step 1). To avoid redundancy, there must not be any rule $q' \in Q_i$ with left hand side $l_{q'} \colon L_q \hookleftarrow T_i$ such that $l'_q = \lambda_i \circ \nu$ and q' is an occurrence of q, i.e. if Σ_{λ_i} maps[55] q' to q.

3. An additional requirement is that $\nu \colon L_q \hookleftarrow T_i$ is *concurrent*. This will be defined below in Definition 5.5. Equivalently, as will be made precise in Proposition 5.6, one could require that there is a derivation $s_i \models\!\!\!\Rightarrow_{Q_i} a$ in the grammar U_i such that ν is covered by a, i.e. the inclusions $\nu \sqsubseteq a$ holds in the subobject poset $\mathrm{Sub}(T_i)$.

4. The morphism ν from step 2 determines the left hand side of a new rule occurrence $\bar{q}$ of q. More detailed, the left leg of $\bar{q}$ is $\alpha_q \colon \nu \to (\nu \circ \alpha_q)$ in $\mathbb{C} \downarrow T_i$, where $L_q \hookleftarrow \nu \rightarrow T_i$ and $K_q \hookleftarrow \nu \circ \alpha_q \rightarrow T_i$ are objects of $\mathbb{C} \downarrow T_i$.

5. It remains to define the right leg of $\bar{q}$; for this purpose, "enlarge" T_i by a "fresh copy" of R_q. This is achieved by taking a pushout $T_i \hookleftarrow t_i \rightarrow T_{i+1} \leftarrow \bar{r} \rightarrowtail R_q$

[55] Given any $\tilde{q} = (l' \leftarrow \alpha' - k' - b' \rightarrow r') \in Q_i$, which is a $\mathbb{C}\downarrow T_i$-rule, the functor $\Sigma_{\lambda_i} \colon \mathbb{C}\downarrow T_i \to \mathbb{C}\downarrow T$ is said to *map* the rule $\tilde{q}$ to the $\mathbb{C}\downarrow T$-rule $\Sigma_{\lambda_i}(\tilde{q}) = (\lambda_i \circ l') \leftarrow \alpha' - (\lambda_i \circ k') - \beta' \rightarrow (\lambda_i \circ r')$.

of $T_i \leftarrow\!\nu\circ\alpha_q\!\rightarrowtail K_q \hookleftarrow\!\beta_q\!\rightarrow R_q$, yielding the pushout square $\begin{smallmatrix} K_q & \rightarrow & R_q \\ T_i & \rightarrow & T_{i+1} \end{smallmatrix}$.

$$\begin{array}{ccc} K_q & \overset{\beta_q}{\hookrightarrow} & R_q \\ {\scriptstyle \nu\circ\alpha_q}\downarrow & & \downarrow{\scriptstyle \bar{r}} \\ T_i & \underset{t_i}{\hookrightarrow} & T_{i+1} \end{array} \qquad \bar{q}: \quad \begin{array}{ccccc} L_q & \overset{\alpha_q}{\leftarrow} & K_q & \overset{\beta_q}{\hookrightarrow} & R_q \\ & {\scriptstyle t_i\circ\nu}\searrow & \downarrow & \swarrow{\scriptstyle \bar{r}} & \\ & & T_{i+1} & & \end{array} \qquad \begin{array}{ccccc} K_q & & \overset{\beta_q}{\hookrightarrow} & & R_q \\ {\scriptstyle \nu\circ\alpha_q}\downarrow & & & & \downarrow{\scriptstyle \bar{r}} \\ T_i & \underset{t_i}{\hookrightarrow} & T_{i+1} & \overset{\lambda_{i+1}}{\dashrightarrow} & T \end{array} \quad \begin{array}{l} {\scriptstyle r_q}\searrow \\ \\ \\ \end{array} \qquad (2)$$

The right hand side of the new rule occurrence $\bar{q}$ is $\bar{r}\colon R_q \hookrightarrow T_{i+1}$. The complete rule occurrence $\bar{q}$ is shown in the middle of (2). Apart from adding $\bar{q}$, it remains to embed the rules of Q_i into T_{i+1}. More precisely, $U_{i+1} = \langle \{\bar{q}\} \cup \Sigma_{t_i}(Q_i), T_{i+1} \rangle$.

Finally, the folding morphism $\lambda_{i+1}\colon T_{i+1} \to T$ is the mediating morphism from the pushout object T_{i+1} as shown in the rightmost diagram in (2). This means that λ_{i+1} is the unique morphism from T_{i+1} to T which satisfies the two equations $\lambda_{i+1} \circ t_i = \lambda_i$ and $\lambda_{i+1} \circ \bar{r} = r_q$ where $r_q\colon R_q \to T$ is the right hand side of the rule q in the original grammar.

Repeated application of these steps yields a sequence of "inclusions"

$$U_0 \text{ “}\subseteq\text{”} U_1 \text{ “}\subseteq\text{”} U_2 \text{ “}\subseteq\text{”} \cdots U_i \text{ “}\subseteq\text{”} U_{i+1} \cdots$$

and in particular a chain $S = T_0 \hookleftarrow\!t_0\!\rightarrow T_1 \hookleftarrow\!t_1\!\rightarrow \cdots T_i \hookleftarrow\!t_i\!\rightarrow T_{i+1} \cdots$ of growing type objects. In the case of strictly finite systems, i.e. those which have a finite transition system without cycles, also this chain is finite and has a largest object T_n and $U_n = \langle Q_n, s_n\colon S \hookrightarrow T_n \rangle$ is the *full* unfolding of G. The object T_n models the totality of all (instances of) "resources" that might be used in some of the possible system runs, and the set Q_n contains one rule for each possible event in the system. Further, rule occurrences that are mapped to the same rule in the original grammar correspond to several different events of the same kind.

Finally, by construction, for each derivation $S \!-\!s\!\rightarrow T \Doteq\!\Rightarrow_Q A \!-\!a\!\rightarrow T$ there is a canonical embedding of A into T_n via some morphism $a'\colon A \hookrightarrow T_n$ such that $a = \lambda_n \circ a'$. More detailed, each derivation $s \Doteq\!\Rightarrow_Q a$ in the original grammar has a uniquely determined counterpart $s_n \Doteq\!\Rightarrow_{Q_n} a'$ in the unfolding U_n. This is made more precise below in Section 5.3, which discusses the *completeness* of unfoldings.

5.3 Occurrence grammars

For a formal treatment of unfoldings, the notions of occurrence grammars (Definition 5.3 below) and concurrent subobjects (Definition 5.5 below) are

still missing. The former will describe the class of all grammars that can be obtained as the result of the unfolding procedure, and the latter will give a static description of all those subobjects of the type object of an occurrence grammar which are coverable by some reachable object.

Occurrence grammars can be seen as a non-deterministic variant of trace processes, and generalize the occurrence graph grammars of [Baldan, 2000]. The difference of trace processes and occurrence grammars is that the latter have only a fixed starting point, the start object, and that in general there are several runs of occurrence grammars that are in conflict. Moreover, occurrence grammars are allowed to be infinite – a fact which will be used in Section 5.4 to construct "full" unfoldings of possibly infinite systems.

☼ Definition 5.3 (Occurrence grammar) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category and $T \in \mathbb{C}$ be an object; let $\mathbb{C}{\downarrow}T$ denote the full subcategory of the slice category $\mathbb{C}\downarrow T$ which has exactly all $\mathscr{M}$-morphisms as objects.

Now an *$\mathscr{M}$-occurrence grammar over T* is defined as a T-typed grammar $O = \langle Q, s\colon S \hookrightarrow T\rangle$ such that s is an $\mathscr{M}$-morphism and Q is a countable set of $\mathbb{C}{\downarrow}T$-rules and moreover the following hold.

causal soundness: The start object has no causes, i.e. $\lfloor s \rfloor = \varnothing$.

causal completeness: All left-hand sides are covered by their causes, i.e. $l_q \sqsubseteq s \sqcup \bigsqcup_{p' \in \lfloor l_q \rfloor} r_{p'}$ holds for all $q \in Q$.

type minimality: The type object is the union of all right hand sides and the start object, i.e. $\mathrm{id}_T \cong s \sqcup \bigsqcup_{q \in Q} r_q$.

backwards determinism: There are no backward conflicts in the causality relation, i.e. $r_q \sqcap r_{q'} \sqsubseteq k_q \sqcup k_{q'}$ for all $q \neq q' \in Q$.

local finiteness: For each rule $q \in Q$, the set of its causes $\lfloor q \rfloor$ is finite.

acyclicity: The transitive-reflexive closure $\preceq$ of the causality relation $\prec$ is a partial order and $\nearrow|_{\lfloor q \rfloor} := \nearrow \cap (\lfloor q \rfloor \times \lfloor q \rfloor)$ is acyclic for each $q \in Q$.

If $\mathbb{C}$ does *not* have $\mathscr{M}$-effective unions, the following must be required explicitly:

dissectability: The join $s \sqcup \bigsqcup_{q \in \lfloor Q' \rfloor} r_q$ exists in $\mathrm{Sub}(T)$ for any finite set $Q' \sqsubseteq Q$ and it is actually an $\mathscr{M}$-subobject, i.e. an element of $\mathrm{Sub}_{\mathscr{M}}(T)$.

☼

Causal soundness and completeness are the same as for trace processes. Type minimality ensures that the type object is just large enough to contain all rules; this corresponds to the fact that the type object of the unfolding exactly models all the (instances of) resources that are possibly used in some run of the system. Backward determinism amounts to the assumption that each event has a unique (minimal) explanation – here and in the sequel, event will often be used as a synonym for a rule in an occurrence grammar instead of the more precise rule occurrence. Local finiteness corresponds to the presupposition that each event may only have a finite number of causes. Acyclicity implies that it is possible to list the finite number of causes of each event such that the resulting order is consistent with causality and asymmetric conflict.

As mentioned before, finite occurrence grammars are exactly those grammars which arise as the result of the unfolding algorithm. The main facts about occurrence grammars – their safety, the coincidence of coverable and concurrent "markings", and their completeness – are discussed next.

Safety of occurrence grammars. A well-known property of occurrence nets is their safety, i.e. each reachable marking of an occurrence net has at most one token in each place. Occurrence grammars enjoy an analogous property, namely that each reachable object is an $\mathscr{M}$-morphism.

◌ Proposition 5.4 (Safety of occurrence grammars) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category; further let $O = \langle Q, s\colon S \hookrightarrow T' \rangle$ be a consuming occurrence grammar over T'.

Then in any derivation $s \models\!\!\!\Rightarrow_Q a$, the $\mathbb{C}\downarrow T'$-object a belongs to the full subcategory $\mathbb{C}\updownarrow T'$, i.e. a is an $\mathscr{M}$-morphism.

Proof sketch. The proof proceeds by induction on the length of the derivation $s \models\!\!\!\Rightarrow_Q a$, which consecutively applies a number of rules

$$s \models\!\!\!\Rightarrow_{q_1} a_1 \models\!\!\!\Rightarrow_{q_2} a_2 \cdots \models\!\!\!\Rightarrow_{q_n} a_n = a.$$

The base case for the empty derivation $n = 0$ follows from $s \in \mathscr{M}$. The following auxiliary fact will be used in the induction step. Each reached object a_i is contained in the join of all the right hand sides of the preceding rules, i.e. $a_i \sqsubseteq s \sqcup \bigsqcup_{j<i} r_j$ where r_j is the right hand side of q_j (see Lemma D.6). The join $s \sqcup \bigsqcup_{j<i} r_j$ is exactly that part of the type object which is needed to "embed" the first $j-1$ derivation steps.

With this additional hypothesis, the proof of the induction step is as follows. The last derivation step $a_{n-1} \models\!\!\!\Rightarrow_q a$ is depicted in the left hand diagram in

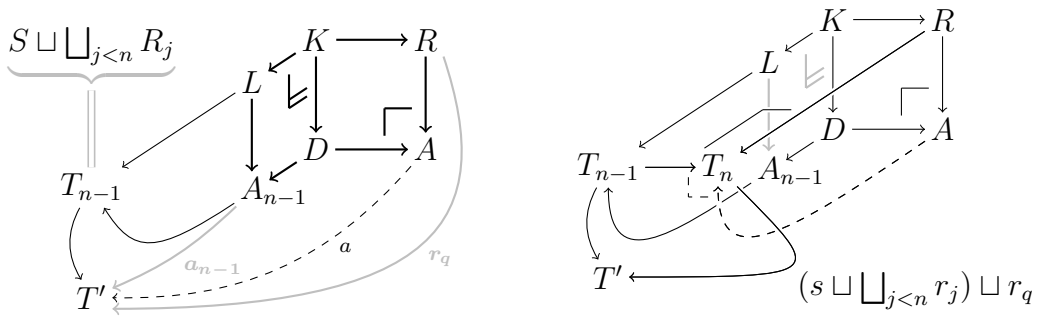


Figure 51: Proof idea for the safety of occurrence grammars

Figure 51; the relevant derivation diagram in $\mathbb{C}$ is marked with thick arrows. The proof obligation is to check that the mediating morphism $a\colon A \to T'$ is actually an $\mathscr{M}$-morphism.

As shown in Appendix D the equation $r_q \sqcap (s \sqcup \bigsqcup_{j<n} r_j) = k_q$ holds; hence the union $(s \sqcup \bigsqcup_{j<n} r_j) \sqcup r_q$ can be computed by taking the pushout of the span $T_{n-1} \leftarrow K \rightarrow R$, yielding a pushout square ${}^{K}_{T_{n-1}}\!\downarrow\!{}^{R}_{T_n}$, which forms the topmost face of the right hand diagram in Figure 51. Now, because the morphism $T_{n-1} \leftarrow K$ factors as $T_{n-1} \leftarrow L \leftarrow K$, there exists a mediating morphism $A \to T_n$ from the pushout ${}^{K}_{D}\!\downarrow\!{}^{R}_{A}$, yielding a second pushout square ${}^{D}_{T_{n-1}}\!\downarrow\!{}^{A}_{T_n}$, i.e. $A \to T_n$ arises by pushout splitting (see LemmaA.14). Since $\mathscr{M}$-morphisms are stable under pushout, this implies that $A \to T_n$ is an $\mathscr{M}$-morphism; however then also the composition $A \to T_n \to T'$ – this is exactly the arrow $a\colon A \to T'$ – is an $\mathscr{M}$-morphism, because $T_n \to T'$ is an $\mathscr{M}$-morphism and the class of $\mathscr{M}$-morphisms is closed w.r.t. composition. □

There are two basic building blocks of this proof: first, it is possible to restrict occurrence grammars to the finite set of rules that are applied in a given derivation as if the applied rules were chosen first in the unfolding algorithm; second, the right hand sides of the applied rules do overlap with the right hand sides of preceding rules only on the preserved part, i.e. the gluing object. Also the second building block reflects how unfoldings are constructed, namely by consecutively adjoining right hand sides of rules via pushout. Indeed, the pushout ${}^{K}_{T_{n-1}}\!\downarrow\!{}^{R}_{T_n}$ in Figure 51 and the pushout ${}^{K_q}_{T_i}\!\downarrow\!{}^{R_q}_{T_{i+1}}$ in (2) can be thought of as two sides of the same coin.

Concurrency and static coverability. In the abstract unfolding algorithm as presented in Section 5.2, the notion of concurrent subobject of an occurrence

grammar was left undefined; concurrent subobjects will turn out to be exactly those subobjects of the type object of a given occurrence grammar which are coverable by some reachable object.

☼ Definition 5.5 (Concurrent subobjects) Let $O = \langle Q, s\colon S \hookleftarrow T' \rangle$ be an occurrence grammar in some weakly $\mathscr{M}$-adhesive category. An $\mathscr{M}$-subobject $[a] \in \mathrm{Sub}_{\mathscr{M}}(T')$ is called *concurrent*

- ⚹ if it has a finite number causes which do not intersect with its co-causes, i.e. $\lfloor a \rfloor$ is finite and $\lfloor a \rfloor \cap \ulcorner a \urcorner = \varnothing$; and
- ⚹ additionally asymmetric conflict on the causes of $[a]$ is linearizable, i.e. the relation $\nearrow|_{\lfloor a \rfloor}$ is acyclic.

☼

As mentioned before, concurrent subobjects coincide with the coverable objects; hence the definition of concurrent subobject as given above provides a *static* counterpart of the notion of coverable object. This is made precise by the following proposition.

◌ Proposition 5.6 (Static coverability) Let $O = \langle Q, s\colon S \hookleftarrow T' \rangle$ be an occurrence grammar, and let $[a] \in \mathrm{Sub}_{\mathscr{M}}(T)$ be an concurrent $\mathscr{M}$-subobject.

Then $[a]$ is concurrent if and only if there is a derivation $s \fatsemi Q \Rightarrow a'$ in O such that $[a']$ covers $[a]$, i.e. such that the inclusion $[a] \sqsubseteq [a']$ holds in $\mathrm{Sub}(T)$.

proof idea. First assume $[a]$ to be concurrent. Then the causes of $\lfloor a \rfloor$ form a so-called *configuration* (see Definition 5.7 below). This implies that there is a linearization $q_1, \ldots, q_n$ of $\lfloor a \rfloor$ and a derivation $s \fatsemi Q \Rightarrow a'$ applying each rule in $\lfloor a \rfloor$ exactly once (see Lemma 5.8 below). That actually the inclusion $[a] \sqsubseteq [a']$ holds is shown by induction on the size of $\lfloor a \rfloor$.

The converse direction is a consequence of Lemma D.6. □

This proposition ensures that no reachability problems have to be solved during the unfolding of grammars. Without this proposition the practical applicability of the general unfolding method would be at least questionable because already in the case of Petri nets with inhibitor arcs the exact unfolding, is too time consuming as there is no suitable notion of concurrent subobject and one has to resort to the dynamic notion of reachable object.

One crucial fact of the proof of Proposition 5.6, is the executability of the set of causes of a given concurrent subobject, i.e. the set $\lfloor a \rfloor$ of $[a] \in \mathrm{Sub}_{\mathscr{M}}(T)$. More generally, in an occurrence grammar $O = \langle Q, s \rangle$, certain "linearizable" sets of rules give rise to derivations by applying its elements in any order that

is compatible with asymmetric conflict; the technical term for such sets of executable rules is *configuration*.

Definition 5.7 (Configuration) Let $O = \langle Q, s\rangle$ be an occurrence grammar. A set of rules $C \subseteq Q$ is a *configuration* if

(i) asymmetric conflict on C does not contain cycles, i.e. $\nearrow|_C$ is acyclic;

(ii) each rule in C has finitely many predecessors w.r.t. asymmetric conflict, i.e. for all $q \in Q$, the set $\{q' \in C \mid q' \nearrow q\}$ is finite;

(iii) the configuration is downward closed w.r.t. causality, i.e. $\lfloor C \rfloor \subseteq C$.

A *compatible linearization* of a configuration $C \subseteq Q$ is a linearization $\{q_i\}_{i\in\{\!|C|\!\}}$ of C which is compatible with asymmetric conflict, i.e. an bijective function $q_\_ \colon \{\!|C|\!\} \xrightarrow{\cong} C$ such that $q_i \nearrow q_j$ implies $i < j$ for all $i, j \in \{\!|C|\!\}$ where $|C|$ is the cardinality of C and $\{\!|C|\!\}$ is the set of natural numbers below $|C|$, i.e. $\{\!|C|\!\} = \{n \in \mathbb{N} \mid n < |C|\}$.[56]

An example of configurations is the set $\lfloor q \rfloor$ where q is a rule of some occurrence grammar. Indeed, $\lfloor q \rfloor$ is the smallest configuration containing q; however it may be possible to enlarge $\lfloor q \rfloor$ by further predecessors $q'' \nearrow^* q$, which means that configurations need not be closed under asymmetric conflict. The set $\{q'' \in Q \mid q'' \nearrow^* q\} \cup \lfloor q \rfloor$ can be a configuration if it is finite and does not contain any cycles of asymmetric conflict. Finally, note that also configurations are defined in static terms.

The statement that each configuration of an occurrence grammar is executable is made precise in the following lemma in terms of compatible linearizations. The lemma also makes explcit the silent assumption that the ambient category behaves well w.r.t. single pushout rewriting, namely each rule should be applicable at any match candidate.

Lemma 5.8 (Executability of configurations) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category such that $\mathrm{Par}(\mathbb{C})$ has pushouts along total $\mathscr{M}$-morphisms, i.e. pushouts along morphisms of the form $[m\rangle$ for some $\mathscr{M}$-morphism $m \in \mathscr{M}$. Further let $O = \langle Q, s\colon S \hookleftarrow T'\rangle$ be an consuming occurrence grammar over T' and let $C \subseteq Q$ be a configuration.

Then

1. there is a compatible linearization $\{q_i\}_{i\in\{\!|C|\!\}}$ of C; and

[56] This includes the case where C is infinite, i.e. $|C| = \infty$, enforcing $n < \infty$ by definition.

2. every compatible linearization $\{q_i\}_{i\in\{\!|C|\!\}}$ of C is executable, i.e. there is a family of $\mathscr{M}$-morphisms $\{a_i\colon A_i \hookrightarrow T'\}_{i\in\{\!|1+|C|\!\}}$ with $a_0 = s$ such that there is a derivation $a_i \models\!\!\!\cdot_{q_i}\!\!\Rightarrow a_{i+1}$ for all $i \in \{\!|C|\!\}$.[57]

In fact, the subobjects that are reached via such linearizations of configurations are all the same as shown in Appendix D. Hence it it possible to define *the* resulting subobject of a configuration. As yet another fact, the latter can be defined as the maximal concurrent subobject with all causes within the configuration and all co-causes outside of the configuration (see Proposition D.8).

☼ Definition 5.9 (Result of configurations) Let $O = \langle Q, s\colon S \hookrightarrow T\rangle$ be an occurrence grammar and $C \subseteq Q$ be a configuration. Then the *result of* C, written $[\![C]\!]$, is

$$[\![C]\!] := \bigsqcup\big\{a \in \mathrm{Sub}_{\mathscr{M}}(T) \mid \lfloor a\rfloor \subseteq C \;\&\; \ulcorner a\urcorner \cap C = \varnothing\big\},$$

which is the greatest subobject $[a] \in \mathrm{Sub}_{\mathscr{M}}(T)$ which satisfies $\lfloor a\rfloor \subseteq C$ and $\ulcorner a\urcorner \cap C = \varnothing$. ☼

Completeness of unfoldings. The third fact about unfoldings is their completeness. Given an unfolding U of some grammar G then each derivation in the original grammar G has a uniquely determined counterpart in the (full) unfolding of G. However the unfolding algorithm described in Section 5.2 terminates only if the grammar G is strictly finite, i.e. if the transition system of G is finite (up to isomorphism).

A more formal statement of completeness is as follows. Given a strictly finite, consuming grammar $G = \langle Q, s\colon S \to T\rangle$, the unfolding algorithm terminates and yields an occurrence grammar $U = \langle Q', s'\colon S \hookrightarrow T'\rangle$ and a folding morphism $\lambda\colon T' \to T$. Then, by construction, the unfolding U is *(fully) complete*: for each derivation sequence

$$s \models\!\!\!\cdot_{\langle q_1, m_1\rangle}\!\!\Rightarrow a_1 \models\!\!\!\cdot_{\langle q_2, m_2\rangle}\!\!\Rightarrow a_2 \cdots \models\!\!\!\cdot_{\langle q_n, m_n\rangle}\!\!\Rightarrow a_n \text{ in } G$$

there are (uniquely) determined rule occurrence $q'_1, \ldots, q'_n \in Q'$ and $\mathscr{M}$-morphisms $a'_1, \ldots, a'_n \in \mathbb{C}\!\updownarrow\! T$ such that there exists a derivation

$$s' \models\!\!\!\cdot_{\langle q'_1, m_1\rangle}\!\!\Rightarrow a'_1 \models\!\!\!\cdot_{\langle q'_2, m_2\rangle}\!\!\Rightarrow a'_2 \cdots \models\!\!\!\cdot_{\langle q'_n, m_n\rangle}\!\!\Rightarrow a'_n$$

[57] Once more, the infinite case is covered by defining $1 + \infty = \infty$.

in the grammar U_i and for all $i \in \{1, \ldots, n\}$, the a_j and the q_j are the images of the a'_j and q'_j w.r.t. the folding morphism λ.[58] This property will be of importance in the proof of the coreflective characterization of unfoldings in Appendix E, and another statement of *full* completeness is given in Lemma E.5.

In the context of state space exploration, where it only matters if some state is reachable and it is immaterial how sates are reached, a weaker notion of completeness is sufficient. It is this weaker notion of completeness that is used in McMillan's finite complete prefix method.

Digression: computing finite complete prefixes. Given a (finite state) system, to verify that a certain state is reachable or to ensure that no reachable state contains certain "bad" substructures, only the set of reachable states needs to be considered, and in particular the derivations which lead to the reachable states are immaterial. Hence, assuming that the given system is described by a grammar $G = \langle Q, s\colon S \hookleftarrow T \rangle$, it might not be necessary to construct the *full* unfolding of G to verify the reachability of a particular state or to check the well-formedness of all states. The unfolding algorithm of Section 5.2 can be stopped if it is clear that each state of the original grammar is accounted for, i.e. if for each derivation $s \models_Q \Rightarrow a$ in G there is some arbitrary derivation $s' \models_{Q'} \Rightarrow a'$ in the unfolding $U = \langle Q', s'\colon S \hookleftarrow T' \rangle$ such that a is the image of a' via the folding morphism.

For the concrete setting of Petri nets with read arcs, a non-trivial effective decision procedure to determine when it is safe to stop the unfolding algorithm was studied in [Baldan et al., 2008e]. The main idea is to impose an extra check in each iteration of the unfolding algorithm which prohibits the addition of a new rule occurrence if there is a "simpler", equivalent way to obtain the "same effect" without the new rule occurrence. Of course, having the same effect is only meaningful if each rule occurrence is considered in the context of a configuration which contains other rule occurrences that can be executed before it; such a context of a rule occurrence has been dubbed *history* in [Baldan et al., 2008e]. With the notions and concepts developed so far, the definition of the latter work can be directly lifted to adhesive rewriting systems.

☼ Definition 5.10 (History) Let $O = \langle Q, s \rangle$ be an occurrence grammar, let $C \subseteq Q$ be a configuration, and let $q \in Q$ be a rule.

[58] This is means that $a_j = \lambda \circ a'_j$ and each rule $q_j = (l_j \leftarrow k_j \rightarrow r_j)$ in the grammar G is equal to $(\lambda \circ l'_j) \leftarrow (\lambda \circ k'_j) \rightarrow (\lambda \circ r'_j)$ where $q'_j = (l'_j \leftarrow k'_j \rightarrow r'_j)$ is the rule occurrence q'_j.

The *history of q w.r.t. C* is the set

$$C(q) := \{q' \in C \mid q' \ (\nearrow|_C)^* \ q\}.$$

A *history of q* is any configuration H which contains q and arises as such a history, i.e. a history is any subset $H \subseteq Q$ such that $H = C(q)$ holds true for some configuation $C \subseteq Q$. ☼

Summarizing, rule occurrences in isolation are replaced by rule occurrences in the context of (a set of) possible histories; a pair of a rule occurrence and one of its histories is a so-called *enriched event* [Baldan et al., 2008e]. As a natural consequence, also occurrence grammars can be enriched with information about the "useful" histories of rule occurrences. The straighforward generalization is as follows.

☼ Definition 5.11 (Enriched occurrence grammar) Let $O = \langle Q, s\rangle$ be an occurrence grammar. A *histories map for O* is a function $\chi\colon Q \to \wp(\wp(Q))$ which maps each rule $q \in Q$ to a nonempty set of histories of q. An *enriched occurrence grammar* is a pair $E = \langle O, \chi\rangle$ where O is an occurrence grammar and χ is a histories map for O. ☼

The requirement that each event has a non-empty set of histories reflects the fact that occurrence grammars should not contain superfluous rules: a rule occurrence with an empty set of histories can be removed from the set of rules.

To determine when a complete prefix of the full unfolding is reached in the unfolding algorithm, it is necessary to store information concerning the "useful" histories of rule occurrences. Each of the histories corresponds to a best way to account for (parts of) reachable states of the original grammar. Hence, during the construction of finite complete prefixes, a transition is added to the unfolding only in case that it is clear that it is part of such a "useful" history.

The above concepts appear to be all that are needed to describe a generalization of the finite complete prefix algorithm of [Baldan et al., 2008e]. The main observation is that the definitions of histories and enriched occurrence grammar only depend on the relations of causality and asymmetric conflict and are independent of the structure of the rewritten objects. Hence, without proof, the claim is that the results of [Baldan et al., 2008e] can be applied to single pushout rewriting in any weakly $\mathscr{M}$-adhesive category.

This concludes the discussion of finite systems. In summary, one might be tempted to believe that most theorems of the theory of graph transformation that only involve finitely many direct derivation diagrams can be lifted to rewriting systems in any (weakly) $\mathscr{M}$-adhesive category. This is of course

not the case, as for example the main theorem of [Bonchi and Heindel, 2006] is only valid in weakly $\mathscr{M}$-adhesive categories with $\mathscr{M}$-effective unions. The question whether the case is different if one relies on the "original" adhesive categories is part of the motivation for the elaborations in the second part of this thesis where also some arguments for a negative answer are provided.

However, in the case of the full unfolding of *infinite* systems, which will be discussed next, the situation is clear. The category in which we are performing the unfolding is required to have additional structure in a completely different sense, namely for every chain of embeddings[59], a (suitably well-behaved) colimit has to exist. This property does not hold in the (adhesive) category of *finite* graphs; however it holds in the category of all *countable* graphs.

5.4 Full unfoldings of possibly infinite systems

The final part of this section is a short overview of full unfoldings of (models of) systems with an *unbounded number* of different evolutions; for these infintite-state systems, there is no hope to obtain finite unfoldings. However, at least on the conceptual level, it is possible to coalesce the growing chain of unfoldings that is generated by the algorithm of Section 5.2 into a single structure. The involved theoretical concept is the *transfinite composition* of chains of (mono-)morphisms (see Definition 5.12 below).

In the context of the unfolding algorithm for Petri nets presented in Section 5.1), McMillan has described the main idea of full unfoldings as follows (see [McMillan, 1995, p. 48]).

> If you make your choices fairly and have infinite time, you will build the unfolding of your Petri net [...].

However, it remains to give a formal counterpart of *fair choices* and to provide a suitable interpretation of what is meant by *having infinite time*.

As for the first point, namely fairness of the choice of rule occurrences, let $G = \langle Q, s\colon S \to T\rangle$ be a grammar, and let

$$U_0 \text{ “}\subseteq\text{” } U_1 \text{ “}\subseteq\text{” } U_2 \text{ “}\subseteq\text{” } \cdots U_i \text{ “}\subseteq\text{” } U_{i+1} \cdots$$

be a chain of (partial) unfoldings that arises by repeated application of the abstract unfolding algorithm in Section 5.2. In this chain, each grammar U_i

[59] *Embedding* is used here instead of *$\mathscr{M}$-morphism of an admissible class of monomorphisms $\mathscr{M}$*.

is of the form $U_i = \langle Q_i, s_i \colon S \hookleftarrow T_i \rangle$ and there is a chain of morphisms $\{t_i \colon T_i \hookleftarrow T_{i+1}\}_{i \in N}$ which embeds each U_i into the next grammar U_{i+1}; finally there are folding morphisms $\lambda_i \colon T_i \to T$ which relate the U_i to G.

Now such a sequence of unfoldings is called *fair* if for each rule $q \in Q$ and all $i \in \mathbb{N}$, every (new) occurrence of q in U_i will be present in some larger U_j for some natural number $j > i$. More precisely, let $(L_q \xrightarrow{l_q} T) \in \mathbb{C} \downarrow T$ be the left hand side of a rule $q \in Q$. Then, this the sequence is fair w.r.t. q, if for all $i \in \mathbb{N}$ and every concurrent subobject of the form $l' \colon L_q \hookleftarrow T_i$ in U_i which satisfies $\lambda_i \circ l' = l_q$, there is a natural number $j \in \mathbb{N}$ and a rule $q_j \in Q_j$ with left hand side l_j such that $l_j = t_i^j \circ l'$ holds where $t_i^j \colon T_i \hookleftarrow T_j$ is the embedding of T_i into T_j, i.e. the morphism t_i^j is the composition of all morphisms of the family $\{t_n \colon T_n \hookleftarrow T_{n+1}\}_{j \leq n < i}$. Then the unfolding is fair if it is fair w.r.t. all rules in Q.

As for the second point, namely the infinitary character of unfoldings, consider the example of Petri nets. Let

$$U_0 \text{ “}\subseteq\text{” } U_1 \text{ “}\subseteq\text{” } U_2 \text{ “}\subseteq\text{” } \cdots U_i \text{ “}\subseteq\text{” } U_{i+1} \cdots$$

be a growing chain of (partial) unfoldings of a Petri net; they can be combined canonically by assembling all the places and transitions that are present in any one of the partial unfoldings U_i into two sets that contain all these places and transitions, respectively. More precisely, if $U_i = \langle P_i, T_i, \mathrm{pre}_i, \mathrm{post}_i \rangle$ is the i-th partial unfolding, then the full unfolding $U = \langle P', T', \mathrm{pre}', \mathrm{post}' \rangle$ would have $P' = \bigcup_{i \in \mathbb{N}} P_i$ and $T' = \bigcup_{i \in \mathbb{N}} T_i$ as the sets of places and transitions, respectively; further, since every $t \in T'$ is contained in T_i for some (smallest) $i \in \mathbb{N}$, the functions pre' and post' are determined by $\mathrm{pre}'(t) = \mathrm{pre}_i(t)$ and $\mathrm{post}'(t) = \mathrm{post}_i(t)$.

However, if only finite sets were allowed, the full unfolding could not be constructed, and it is trivial that full unfoldings cannot be computed effectively in general. Hence, in general, the study of full unfoldings has to go beyond finitistic universes; otherwise it would be impossible to have a single occurrence net which represents all possible evolutions of a given Petri net.

To arrive at a categorical generalization of the union $\bigcup_{i \in \mathbb{N}} X_i$ of a chain of growing sets $X_i \subseteq X_{i+1}$, the following two observations are central: first, the rôle of the inclusions $X_i \subseteq X_{i+1}$ can be played by monomorphisms $X_i \hookleftarrow_{m_i} X_{i+1}$ (of a class $\mathscr{M}$), and second, the set $\bigcup_{i \in \mathbb{N}} X_i$ corresponds to the colimit object which is obtained by taking the colimit of the diagram $\{X_i \hookleftarrow_{m_i} X_{i+1}\}_{i \in \mathbb{N}}$; however, the second step implicitly assumes that chains of monomorphisms (in $\mathscr{M}$) are stable under composition.

☼ Definition 5.12 (ω-composition) Let $\mathbb{C}$ be a category. A $\mathbb{C}$-diagram of the form $\mathcal{X} = \{f_i \colon X_i \to X_{i+1}\}_{i\in\mathbb{N}}$ is referred to as an *ω-chain* of morphisms. Further, an ω-chain $\mathcal{X} = \{f_i \colon X_i \to X_{i+1}\}_{i\in\mathbb{N}}$ is *composable* if there exists a colimit $\langle X, \{x_i \colon X_i \to X\}_{i\in\mathbb{N}}\rangle$ of $\mathcal{X}$ in $\mathbb{C}$, and then the first coprojection $x_0 \colon X_0 \to X$ is *a composition* of the chain $\mathcal{X}$.

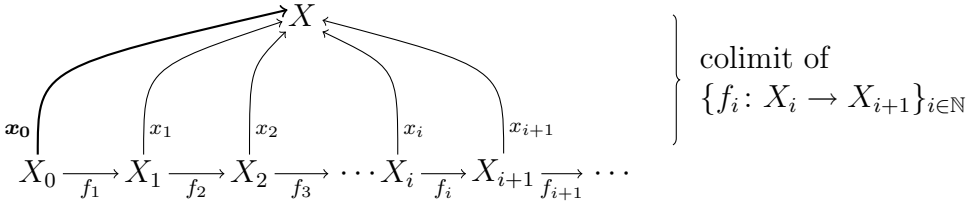


Figure 52: Illustration of the composition of ω-chains of morphism via colimits

Let $\mathscr{M}$ be a class of morphisms that is closed under (binary) composition, which means that $m \circ n \in \mathscr{M}$ holds for all $m, n \in \mathscr{M}$. Then the morphism class $\mathscr{M}$ is *closed under ω-composition* if the following two hold:

composability: ω-chains of $\mathscr{M}$-morphisms are composable, i.e. $\mathbb{C}$ has colimits of diagrams of the form $\mathcal{X} = \{X_i \hookleftarrow_{m_i} X_{i+1}\}_{i\in\mathbb{N}}$, and

closure: compositions of ω-chains of $\mathscr{M}$-morphisms are again $\mathscr{M}$-morphisms, i.e. for all colimits $\langle X, \{x_i \colon X_i \to X\}_{i\in\mathbb{N}}\rangle$ of $\mathcal{X}$, the first coprojection $x_0 \colon X_0 \hookleftarrow X$ belongs to the morphism class $\mathscr{M}$.[60]

☼

If all the morphisms of a chain are set inclusions $X_i \hookrightarrow X_{i+1}$, the canonical composition of the chain $\{X_i \hookrightarrow X_{i+1}\}_{i\in\mathbb{N}}$ is the inclusion $X_0 \hookrightarrow \bigcup_{i\in\mathbb{N}} X_i$. Both composability and closure will be satisfied by the class $\mathscr{M}$ of the infinitary generalizations of weakly $\mathscr{M}$-adhesive categories proposed below in Definition 5.14. That closure of $\mathscr{M}$-morphisms under transfinite compositions is a non-trivial property is witnessed by the following "artificial" example, which is illustrated in Figure 53.

❀ Example 5.13 Consider the following abstract category $\mathbb{X}$ with $\mathrm{ob}(\mathbb{X}) = \mathbb{N} \cup \{\Omega, Z\}$, and the homsets $\mathbb{X}(A, B)$ for $A \neq B \in \mathrm{ob}(\mathbb{X})$ are given by

$$\mathbb{X}(A,B) = \begin{cases} \{\leq\} & \text{if } A \in \mathbb{N} \text{ or } B = \Omega \\ \{x_B, y_B\} & \text{otherwise.} \end{cases}$$

[60] As an immediate consequence, all coprojections must belong to $\mathscr{M}$.

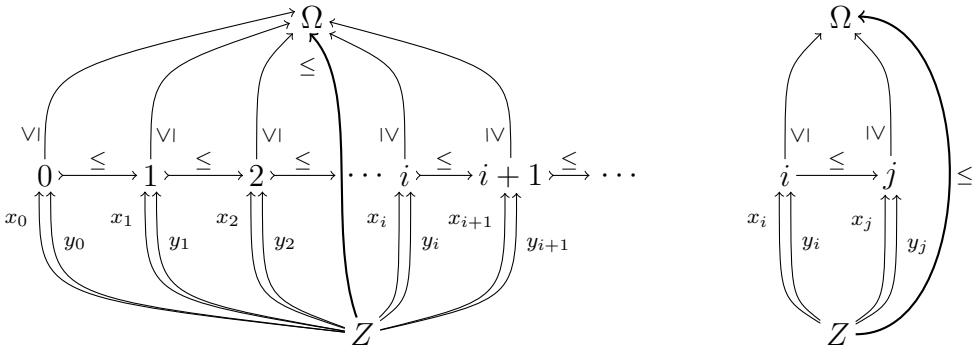


Figure 53: Failure of closure of $\mathscr{M}$-morphisms under ω-composition

Composition is fully determined by the two equations $\leq \circ x_n = x_m$ and $\leq \circ y_n = y_m$ for all $n \leq m \in \mathbb{N}$. Now, the chain $\{i \mathrel{-\!\!\leq\!\!\to} i+1\}_{i\in\mathbb{N}}$ is a chain of monomorphisms and $0 \mathrel{-\!\!\leq\!\!\to} \Omega$ is its transfinite composition; however the latter arrow is not monic, as $x_0 \neq y_0$ while at the same time $\leq \circ x_0 = \leq = \leq \circ y_0$.[61]

However, by requiring that colimits of chains of $\mathscr{M}$-morphisms satisfy the same properties as pushouts of pairs of $\mathscr{M}$-morphisms in a weakly $\mathscr{M}$-adhesive category do, one obtains a suitable category which allows the construction of full unfoldings. The formal definition of the resulting class of categories is as follows.

Definition 5.14 (Weakly $\mathscr{M}\omega$-adhesive categories) Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$. Then the category $\mathbb{C}$ is *weakly $\mathscr{M}\omega$-adhesive* if it is weakly $\mathscr{M}$-adhesive and moreover colimits of ω-chains $\{m_i \colon A_i \hookrightarrow A_{i+1}\}_{i\in\mathbb{N}}$ of $\mathscr{M}$-morphisms exist, are preserved by the graphing functor $\Gamma \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C}, \mathscr{M})$ and are universal (see Definition 9.3).

Remark 5.15 (Composability of ω-chains) As shown above, in general, ω-chains of $\mathscr{M}$-morphisms may not compose. However, it is not necessary to include this requirement in the above definition as it is a consequence of it (see Lemma B.6).

Now, working in a weakly $\mathscr{M}\omega$-adhesive category the full unfolding of a fair unfolding chain can be obtained as follows. Let $G = \langle Q, s \colon S \to T \rangle$ be a

[61] The interested reader might want to check whether the category of the present example is actually $\mathscr{M}$-adhesive where $\mathscr{M}$ is the class of all monomorphisms between two natural numbers.

consuming, $\mathscr{M}$-linear grammar typed over $T \in \mathbb{C}$. Further let

$$U_0 \text{ “}\subseteq\text{” } U_1 \text{ “}\subseteq\text{” } U_2 \text{ “}\subseteq\text{” } \cdots U_i \text{ “}\subseteq\text{” } U_{i+1} \cdots$$

be a fair chain of unfoldings that is generated by repeated execution of the unfolding algorithm in Section 5.2 where each grammar $U_i = \langle Q_i, s_i \colon S \hookleftarrow T_i \rangle$ is embedded into the next grammar $U_{i+1} = \langle Q_{i+1}, s_{i+1} \colon S \hookleftarrow T_{i+1} \rangle$ via a morphism $t_i \colon T_i \hookleftarrow T_{i+1}$. Then this chain can be coalesced into a single occurrence grammar as follows.

1. Take some ω-composition $t_0^\infty \colon T_0 \to T^\infty$ of the t_i, which is achieved by taking a colimit $\langle T^\infty, \{t_i^\infty \colon T_i \hookleftarrow T^\infty\}_{i \in \mathbb{N}} \rangle$ of the chain $\{t_i \colon T_i \to T_{i+1}\}_{i \in \mathbb{N}}$.

2. The new start object is $t_0^\infty \colon S \to T^\infty$ since by construction $S = T_0$.

3. The “union” of the Q_i is given by $Q^\infty = \bigcup_{i \in \mathbb{N}} \Sigma_{t_i^\infty}(Q_i)$ where each rule set $\Sigma_{t_i^\infty}(Q_i)$ is the image of the rule set Q_i in the larger type object T^∞; formally $\Sigma_{t_i^\infty}(Q_i) = \{\Sigma_{t_i^\infty}(q) \mid q \in Q_i\}$.

4. Finally, the folding morphism $\lambda^\infty \colon T^\infty \to T$ is the unique mediating morphism $\lambda^\infty \colon T^\infty \to T$ which satisfies $\lambda_i = \lambda^\infty \circ t_i^\infty$ for all $i \in \mathbb{N}$ where $\lambda_i \colon T_i \to T$ is the folding morphism of the grammar U_i.

After these steps, the resulting occurrence grammar $U = \langle Q^\infty, s^\infty \colon S \to T^\infty \rangle$ together with the folding morphism $\lambda^\infty \colon T^\infty \to T$ is the *full unfolding of* G.

This concludes the discussion of the algorithmic aspects of unfoldings of grammars; the omitted details, including the existence of fair unfolding sequences, are given in Appendix E. Next, in contrast to this “constructive” approach to unfoldings, the following section disscusses an alternative declarative characterization of the unfolding procedure. It will turn out that unfolding is not an ad hoc method but that it is actually the universal method to transform a given grammar into an occurrence grammar.

Unfolding as a co-free construction

6

The first fundamental idea of the characterization of the unfolding algorithm as a co-free construction is to shift the focus from the analysis of system models in isolation to the study of their interrelations. In other words, the idea is to study the *social life*[62] of system models. To this end, system models are congregated into categories. An object of a category of this kind is a system model, and the morphisms between two system models typically carry enough information to determine how one system can simulate the other one; more precisely a morphism from one system to another one provides data which allow to infer how the latter system can or should simulate the former. Moreover, synchronization and composition of systems usually can be understood as instances of certain basic universal constructions in a suitable category of system models.

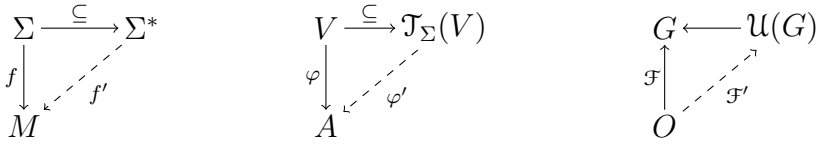
For example, take labeled transition systems as objects and all *simulation strategies* between them as morphisms, where a simulation strategy is a function which maps transitions of the domain system to transitions of the codomain system. To obtain a category, take the trivial simulation strategies as identities (they map each transition to itself); moreover the composition of two strategies is just the composition of the respective functions. Now, the product in this category coincides with the usual product of transition systems (up to isomorphism). However, the product operation also applies to morphisms, which means that also a pair of simulation strategies can be combined to obtain a single simulation between the respective products of the involved systems.

Several different proposals for categories of transition systems and Petri nets exist (see e.g. [Sassone et al., 1996, Hayman and Winskel, 2008]), and also for the special case of graph grammars, more than one notion of grammar morphism has been developed [Baldan, 2000, Ribeiro, 1996]. However, the question concerning "the right" notion of morphism between system models is often only meaningful relative to a given purpose.

In this thesis, the purpose relative to which a suitable notion of grammar morphism is proposed is the characterization of the unfolding algorithm as the universal method to convert a grammar into an occurrence grammar (while preserving the *morphisms as simulations* paradigm). In more technical terms, the goal is to establish that unfolding is a *(co-)free* construction: in the same

[62] The introduction of [Fiadeiro, 2005] presents category theory as a tool for the study of the *social life of objects*.

way as the monoid of words forms the *free* monoid over an alphabet and the term algebra over a signature arises as the *free* algebra, the unfolding of a grammar will be established as the *(co-)free* occurrence grammar. The



(a) Words as the free monoid

(b) Terms as the free algebra

(c) The unfolding as the co-free occurrence grammar

Figure 54: Free and co-free constructions

meaning of these statements is expressed diagrammatically in Figure 54; these diagrams are meant to be read as follows.

In figure Figure 54(a), the set Σ is an alphabet, i.e. a set of letters, and Σ^* is used as shorthand for the monoid of words over the alphabet Σ. That Σ^* is the *free* monoid means that for any function $f\colon \Sigma \to M$ from the alphabet Σ to the underlying set of any monoid $\langle M, e, \cdot\rangle$, there exists a unique monoid homomorphism $f'\colon \Sigma^* \to M$ which coincides with f on single letters.

The same situation can be found in the category of algebras and algebra homomorphisms as depicted in Figure 54(b). The signature Σ contains function symbols, V is a set of variables, and $\mathfrak{T}_\Sigma(V)$ is the term algebra over the signature Σ with variables in V. That it is the *free* such algebra means that for each function $\varphi\colon V \to A$ which maps each variable $v \in V$ to some element of the carrier set of a given Σ-algebra, there exists a unique Σ-algebra homomorphism $\varphi'\colon \mathfrak{T}_\Sigma(V) \to A$ which coincides with φ on variables.

The main result of (the first chapter of) this thesis is that occurrence grammars can be characterized in essentially the same way. The only difference is that the arrows need to be reversed, i.e. unfoldings are "dually free" or *co-free*. Postponing the discussion of the issue of finding a suitable notion of grammar morphism, Figure 54(c) shows what it means that the unfolding $\mathcal{U}(G)$ of some grammar G with folding morphism $\mathcal{U}(G) \to G$ is co-free. Namely for any occurrence grammar O and any grammar morphism $\mathcal{F}\colon O \to G$, there is a unique grammar morphism $\mathcal{F}'\colon O \to \mathcal{U}(G)$ such that $\mathcal{F}'$ is "folded back" to $\mathcal{F}$ and in particular, the simulation strategy described by $\mathcal{F}$ is the chaining of the simulation strategies given by $\mathcal{F}'$ and then folding morphism $\mathcal{U}(G) \to G$. Using the language of category theory this fact can be expressed in one sentence:

the category of occurrence grammars forms a coreflective subcategory of the category of grammars.

Using standard results of category theory, this characterization of the unfolding as a co-free construction gives actually an *unfolding functor*, i.e. a homomorphism between categories, from the category of grammars to the category of occurrence grammars. This unfolding functor does not only act on objects, but also on morphisms which describe the relations between the grammars, i.e. each morphism between grammars is mapped to a corresponding one between the unfoldings of the grammars. Hence the unfolding of grammars is an instance of so-called *functorial semantics*[63] (see e.g. [Bruni et al., 2001]).

A second consequence of the co-freeness of the unfolding construction is the fact that the unfolding functor forms part of a *coreflection*, which is the category theoretic generalization of a Galois embedding. Using standard result of category theory once more, this means that the unfolding functor preserves all limits that exist in the category of grammars, and in particular it preserves any product of grammars. For example, to compute the unfolding $\mathcal{U}(G \times H)$ of the product $G \times H$ of two grammars G and H, it is enough to compute the unfoldings $\mathcal{U}(G)$ and $\mathcal{U}(H)$ and to take the product $\mathcal{U}(G) \times \mathcal{U}(H)$ afterwards since $\mathcal{U}(G) \times \mathcal{U}(H) \cong \mathcal{U}(G \times H)$. Computing the unfolding in this way is usually much more efficient than computing the unfolding $\mathcal{U}(G \times H)$ starting from the product grammar $G \times H$ (see [Baldan et al., 2006b] for more information on such *distributed* unfoldings of systems).

Overview of the section. The major part of this section is devoted to the notion of grammar morphism. Particular attention is given to the *morphisms as simulations* paradigm and it is made explicit in (the proof of) Lemma 6.4 how the proposed grammar morphisms induce simulation strategies between grammars. The main result is stated in Theorem 6.6, which generalizes and improves upon previous characterizations of unfolding as a co-free construction in the area of graph transformation (see e.g. [Baldan, 2000]) and the theory of Petri nets (see e.g. [Nielsen et al., 1981]).

The main difficulties to obtain this result are closely related to the fact that objects of adhesive categories such as graphs have non-trivial symmetries, i.e.

[63] However this functorial semantics is not directly related to the functorial semantics in the sense of [Lawvere, 1963]. The latter approach provides an equivalence with categories of algebras over signatures and algebra homomorphisms on the one hand, and product preserving functors from the Lawvere theories of signatures and natural transformations between them on the other hand.

there are in general several different isomorphisms from an object to itself. The central property of the proposed notion of grammar morphism is its capacity to keep track of these symmetries. The details can be found in Appendix E.

Grammar morphisms as simulation morphisms. As argued above, one essential building block of the characterization of the unfolding as a co-free construction is a suitable notion of morphism between grammars that allows to obtain a sufficiently rich category of grammars. In particular, the morphisms should fit the *morphisms as simulations* paradigm. In the special case of grammars, this means that a morphism from one grammar to another one should preserve derivation steps, i.e. derivation diagrams in the domain grammar should correspond to derivation diagrams in the codomain grammar. Without this requirement, it would also be allowed to have no morphisms between any two (different) grammars, which amounts to "falling back" to the study of grammars in isolation.

Inspecting the simulation requirement for grammar morphisms in more detail, the first condition is that each reachable object in the domain grammar of a morphism should be mapped to a reachable object in the codomain grammar. This is in analogy to the situation for Petri nets, where net morphisms map reachable markings of the domain net to reachable markings of the codomain net. In the Petri net case this is done via multi-relations between the places of the nets (see [Winskel, 1985]). On the abstract level, these multi-relations are replaced by suitable functors (see also the discussion below after Definition 6.2).

☼ Definition 6.1 (Simulation morphism) Let $\mathbb{E}$, $\mathbb{D}$ be categories, and let $\mathcal{F}\colon \mathbb{E} \to \mathbb{D}$ be a functor. Given an $\mathbb{E}$-rule $q = (L \leftarrow\!\alpha\!- K -\!\beta\!\rightarrow R)$, the *image*[64] *of* q *under* $\mathcal{F}$ is the rule

$$\mathcal{F}(q) := \left(\mathcal{F}(L) \xleftarrow{\mathcal{F}(\alpha)} \mathcal{F}(K) \xrightarrow{\mathcal{F}(\beta)} \mathcal{F}(R)\right).$$

Now $\mathcal{F}$ is an (SPO) *simulation morphism* if it preserves SPO derivation steps, i.e. for all $\mathbb{E}$-rules q, if $\mathcal{X}\colon A \models\!\!\!\!\!\!\cdot\;^{\langle q,m\rangle}\!\!\Rightarrow B$ is a direct derivation diagram in $\mathbb{E}$, then $\mathcal{F} \circ \mathcal{X}\colon \mathcal{F}(A) \models\!\!\!\!\!\!\cdot\;^{\langle \mathcal{F}(q),\mathcal{F}(m)\rangle}\!\!\Rightarrow \mathcal{F}(B)$ is a direct derivation diagram in $\mathbb{D}$ (see also Figure 55). ☼

In the context of this thesis, the relevant examples of simulation morphisms have slice categories as domain and codomain, i.e. the categories $\mathbb{E}$ and $\mathbb{D}$ in

[64] This set theoretic terminology is not to be confused with the notion of the image known from factorization systems. Further, one could just define $\mathcal{F}(q)$ as $\mathcal{F} \circ q$, where a rule q is considered as a $\mathbb{C}$-diagram $q\colon (\cdot \leftarrow \cdot \rightarrow \cdot) \to \mathbb{C}$.

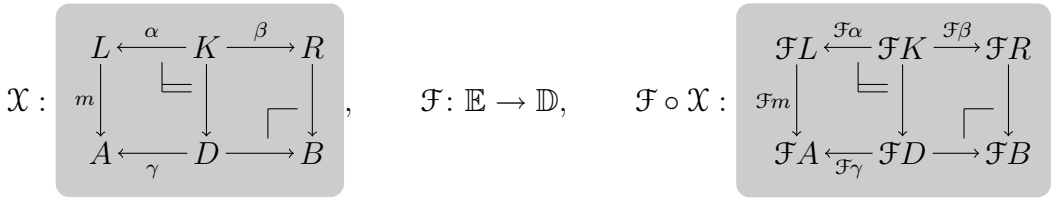
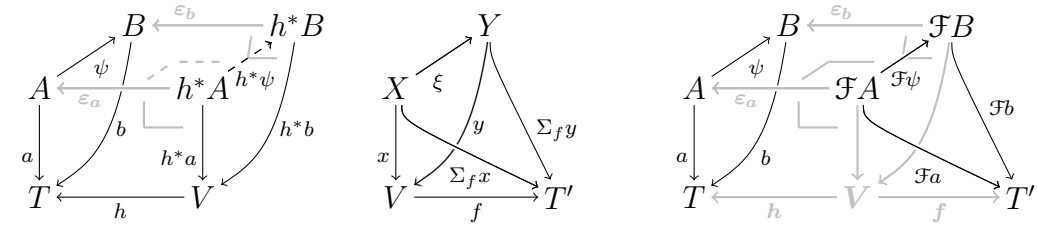


Figure 55: Simulation morphisms via derivation preserving functors

the previous definition will be of the form $\mathbb{C}{\downarrow}T$ and $\mathbb{C}{\downarrow}T'$, respectively, where $T, T' \in \mathbb{C}$ are type objects of grammars. The obvious reason for this is that the unfolding takes typed grammars as input, and that the morphisms between typed grammars will be (certain classes of) structure preserving simulation morphisms between the slice categories over the respective type objects.

In fact, to obtain an analogy with the theory of net morphisms, it is not even necessary to consider all simulation morphisms between slice categories for the definition of grammar morphisms, and may restrict attention the multi-relation-like *retyping functors*.

☼ Definition 6.2 (Pullback and retyping functors) Let $\mathbb{C}$ be a category and $T, V, T' \in \mathbb{C}$ be objects. A functor $\mathcal{H}\colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}V$ is a *pullback functor* if it acts by pulling back along some morphism $T \leftarrow h - V$ as illustrated in Figure 56(a); such a pullback functor is usually written $h^*\colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}V$.



(a) Pullback functor $h^*\colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}V$

(b) Composition with f $\Sigma_f\colon \mathbb{C}{\downarrow}V \to \mathbb{C}{\downarrow}T'$

(c) Retyping functor $\mathcal{F} = \Sigma_f \circ h^*\colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}T'$

Figure 56: The components of a retyping functor

More precisely, $\mathcal{H}$ is a pullback functor if there exists some morphism $h\colon V \to T$ such that $\mathcal{H}$ is a right adjoint to $\Sigma_h\colon \mathbb{C}{\downarrow}V \to \mathbb{C}{\downarrow}T$, in signs $\Sigma_h \dashv h^*$, where Σ_h is the "post-composition with h" functor (see also Figure 56(b)).

Further, a functor $\mathcal{F}\colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}T'$ is a *retyping functor* if there exists a $\mathbb{C}$-span $T \leftarrow h - V - f \rightarrow T'$ such that $\mathcal{F}$ acts by pulling back along h followed by

composition with f, i.e. if $\mathcal{F} = \Sigma_f \circ h^*$; the span $T \leftarrow h - V - f \rightarrow T'$ is called the *underlying span* of $\mathcal{F}$. ☼

The terminology is based on the notion of retyping of [Baldan, 2000]. The proximity to multi-relations is via the underlying span of retyping functors (see [Bruni and Gadducci, 2003] for a study of the relationships between spans and multi-relations).

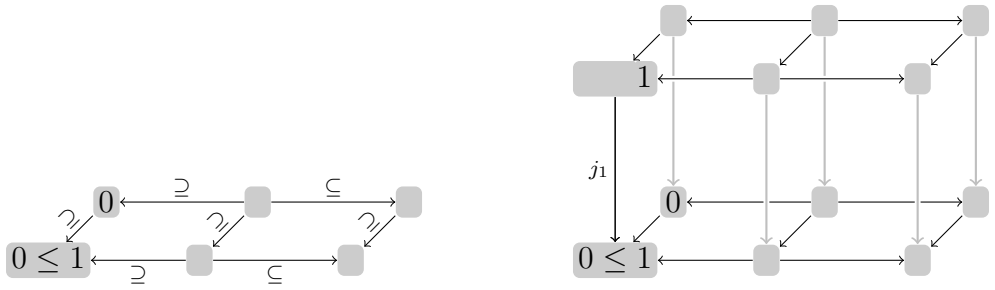
The main technical advantage of retyping functors besides these connections to existing concepts of the established theory of Petri nets is the fact that they are simulation morphisms – at least in most application relevant cases. However, the general case of weakly $\mathscr{M}$-adhesive categories is slightly more involved. The arising issues are illustrated in the following example in the category of preorders.

☙ Example 6.3 (Failure of simulation) Consider the weakly $\mathscr{C}losed$-adhesive category $\mathbb{P}\mathtt{re}\mathbb{O}\mathtt{rd}$ with preorders as objects and order preserving functions as morphisms. An object is any preorder $\langle P, \leq \rangle$, i.e. any set P that comes equipped with a reflexive and transitive relation $\leq \subseteq P \times P$; further, a morphism $f \colon \langle P, \leq \rangle \to \langle P', \trianglelefteq \rangle$ is any order preserving function $f \colon P \to P'$ in $\mathbb{S}\mathtt{et}$, which means that $p_1 \leq p_2$ implies $f(p_1) \trianglelefteq f(p_2)$ for all $p_1, p_2 \in P$.

Further, a subset $X \subseteq P$ of a preorder $\langle P, \leq \rangle$ is *downward-closed* (or *closed*) if $p \leq x$ implies $p \in X$ for all $p \in P$ and all $x \in X$. Now, a morphism $f \colon \langle P, \leq \rangle \to \langle P', \trianglelefteq \rangle$ is *closed* if it preserves downward-closed subsets, i.e. whenever $X \subseteq P$ is downward-closed then also $f(X) = \{f(x) \mid x \in X\} \subseteq P'$ is downward-closed (w.r.t. $\trianglelefteq$). Finally, the class $\mathscr{C}losed$ consists of all injective, closed morphisms, and its elements are also called *closed embeddings*.

Now consider the rule $q = \{0\} \hookleftarrow \varnothing \hookrightarrow \varnothing$, which destroys one element, and take the injection $i_0 \colon \{0\} \hookleftarrow\!\shortmid\!\rightarrow \{0 \leq 1\}$ as match; the type object is $\{0 \leq 1\}$, and the previous preorders are taken implicitly as sub-preorders of it, i.e. as objects of $\mathbb{P}\mathtt{re}\mathbb{O}\mathtt{rd} \downarrow \{0 \leq 1\}$. The result of rewriting and the rewriting step $\{0 \leq 1\} \rightleftharpoons_{\langle q, i_0 \rangle} \varnothing$ in $\mathbb{P}\mathtt{re}\mathbb{O}\mathtt{rd} \downarrow \{0 \leq 1\}$ are illustrated in Figure 57(a). Note that the element 1 must be deleted as well because the injection $i_1 \colon \{1\} \succ\!\shortmid\!\rightarrow \{0 \leq 1\}$ is not closed and pushouts are taken in the category of $\mathscr{C}losed$-partial maps. Hence, in general, application of the rule q does not only delete a single element but also the elements "above" it.

However, as illustrated in Figure 57(b), the above rewriting step is not stable under pullback along the injection $j_1 \colon \{1\} \hookrightarrow \{0 \leq 1\}$, i.e. the retyping functor $j_1^* \colon \mathbb{P}\mathtt{re}\mathbb{O}\mathtt{rd} \downarrow \{0 \leq 1\} \to \mathbb{P}\mathtt{re}\mathbb{O}\mathtt{rd} \downarrow \{1\}$ is not a simulation morphism. The reasons are the following. First, the image of q under j_1^* is the identity on $\varnothing$, i.e. $j_1^*(q) = \varnothing \hookleftarrow \varnothing \hookrightarrow \varnothing$, and hence the application of the latter rule does



(a) Rewriting step $\{0 \leq 1\} \models\!\!\!\!\cdot\langle q, i_0\rangle\!\!\!\!\Rightarrow \varnothing$

(b) The image of the step under the retyping functor j_1^* where $j_1 \colon \{1\} \hookrightarrow \{0 \leq 1\}$

Figure 57: Failure of simulations in the category of preorders

not change anything. Second the image of $\{0 \leq 1\}$ under j_1^* is $\{1\}$, and hence applying $j_1^*(q)$ to $\{1\}$ would yield $\{1\}$ again, i.e. $\{1\} \models\!\!\!\!\cdot_{j_1^*(q)}\!\!\!\!\Rightarrow \{1\}$. However, the image of the rewriting step $\{0 \leq 1\} \models\!\!\!\!\cdot\langle q, i_0\rangle\!\!\!\!\Rightarrow \varnothing$ under j_1^* yields only the "pseudo-step" $\{1\}$ "$\models\!\!\!\!\cdot\!\!\!\!\Rightarrow$" $\varnothing$, which is not an SPO rewriting step.

One solution to this problem, which works in every weakly $\mathscr{M}$-adhesive category, is the restriction to those retyping functors $\mathcal{F} = \Sigma_f \circ n^* \colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}T'$ which are based on spans $T \leftarrow n \rightarrowtail V \xrightarrow{f} T'$ with an $\mathscr{M}$-morphism n as left leg. Hence, the failure of simulation in Example 6.3 is related to the fact that the injection $j_1 \colon \{1\} \hookrightarrow \{0 \leq 1\}$ is not a closed embedding.

Lemma 6.4 (Simulation Lemma) Let $\mathbb{C}$ be any weakly $\mathscr{M}$-adhesive category, and $T \leftarrow n \rightarrowtail V \xrightarrow{f} T'$ be a span with an $\mathscr{M}$-morphism n. Then the retyping functor $\Sigma_f \circ n^* \colon \mathbb{C}\downarrow T \to \mathbb{C}\downarrow T'$ is an SPO simulation morphism.

The proof idea of the Simulation Lemma is illustrated in Figure 58. Given a $\mathbb{C}{\downarrow}T$-rule $q = (l \leftarrow \alpha - k - \beta \rightarrow r)$ and a retyping functor $\mathcal{F} = \Sigma_f \circ n^* \colon \mathbb{C}{\downarrow}T \to \mathbb{C}{\downarrow}T'$, the rewriting step $a \models\!\!\!\!\cdot\langle q, m\rangle\!\!\!\!\Rightarrow b$ is mapped to $\mathcal{F}(a) \models\!\!\!\!\cdot\langle \mathcal{F}(q), \mathcal{F}(m)\rangle\!\!\!\!\Rightarrow \mathcal{F}(b)$. The two upper neighbouring cuboid diagrams consist of pullback squares where $\varepsilon \colon \Sigma_n \circ n^* \to \mathrm{id}_{\mathbb{C}\downarrow T}$ is the counit of the adjunction $\Sigma_n \dashv n^*$. The reason why the rewriting diagram ${}^{L}_{A}\!\downarrow\!\overleftarrow{\square}\!\downarrow\!\overrightarrow{\square}\!\downarrow{}^{R}_{B}$ (over T) gives rise to a rewriting diagram ${}^{\mathcal{F}L}_{\mathcal{F}A}\!\downarrow\!\overleftarrow{\square}\!\downarrow\!\overrightarrow{\square}\!\downarrow{}^{\mathcal{F}R}_{\mathcal{F}B}$ (over T') is that $\mathscr{M}$-final pullback complements and pushouts of pairs of $\mathscr{M}$-morphisms are stable under pullback along $\mathscr{M}$-morphisms.

Though the Simulation Lemma might appear restrictive, it actually ensures that – in any weakly $\mathscr{M}$-adhesive category – there are "enough" retyping functors which are simulation morphisms to obtain the coreflective characterization of the unfolding. Moreover, in most application relevant categories, *all*

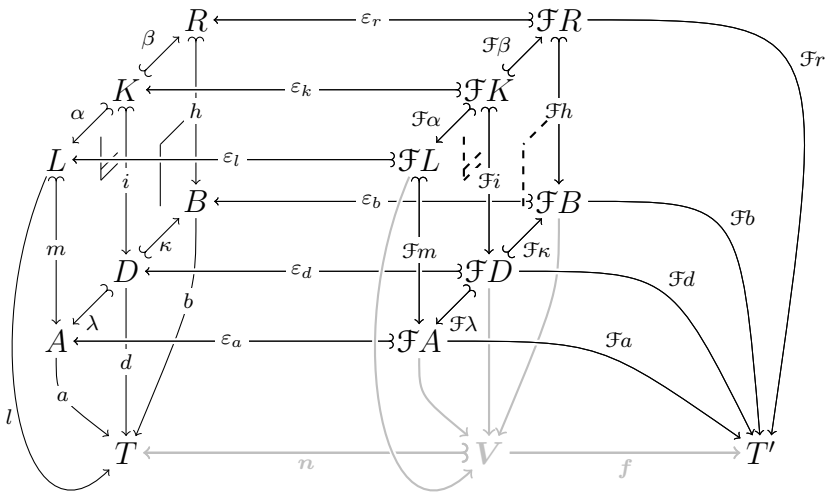


Figure 58: Proof idea of the Simulation Lemma where $\mathcal{F} = \Sigma_f \circ n^*$

retyping functors are simulation morphisms. The justification of this statement is that (concrete) categories often allow for a suitable notion of *image* of morphisms, which is the category theoretic generalization of the set-theoretic image $f(A) = \{f(a) \mid a \in A\} \subseteq B$ of an arbitrary function $f\colon A \to B$. More precisely, it is often the case that each morphism $A \xrightarrow{f} B$ in $\mathbb{C}$ factors as $A \xrightarrow{e} f(A) \xleftarrow{f'} B$ for some $\mathscr{M}$-morphism f' and this factorization is (essentially) unique and stable under pullback (along $\mathscr{M}$-morphisms). Technically speaking, all retyping functors are simulation morphisms in a given weakly $\mathscr{M}$-adhesive category $\mathbb{C}$ if the latter comes equipped with an $(\mathscr{E}, \mathscr{M})$-factorization system such that the class $\mathscr{E}$ is stable under pullback along $\mathscr{M}$-morphisms. This property is closely related to the Frobenius-Reciprocity law as discussed in [Clementino et al., 1996].

The definition of grammar morphism. Independent of which one of the above two solutions one might want to choose, it is safe to assume that there are "enough" retyping functors which are simulation morphisms.[65] Besides this "technical" soundness property of retyping functors, the two main "concep-

[65] In fact, theoretically, it would be enough to consider only functors of the form Σ_f. However, then not even the product of two graph grammars would exist in the category of `Graph` grammars.

tual" observations that lead to the definition of grammar morphism in this thesis are the following two: first, as mentioned e.g. in [Meseguer et al., 1997], multi-relations are nothing else but *monoid homomorphisms* between the free commutative monoids of Petri net markings; second, each *monoid homomorphism* is a special kind of a monotone mapping between the posets of markings. Thus, monoid homomorphisms are a particular example of *category homomorphism*, i.e. *functor*, since each poset can be seen as a particular type of category.

The second observation is related to considerations of [Baldan, 2000] which establish an analogy between net markings of a net with place set P on the one hand and P-colored sets on the other hand (see Example 2.4); i.e. the role of net markings over a place set P is played by objects of the slice category $\mathbb{Set}\downarrow P$. In general, the abstract counterpart of net markings in some typed grammar $G = \langle Q, s\colon S \to T\rangle$ are the objects of the slice category $\mathbb{C}\downarrow T$. Now, a grammar morphisms between grammars $G = \langle Q, s\colon S \to T\rangle$ and $G' = \langle Q', s'\colon S' \to T'\rangle$ is a retyping functor which preserves the structure of grammars.

☼ Definition 6.5 (Grammar Morphism) Let $G = \langle Q, s\colon S \to T\rangle$ and $G' = \langle Q', s'\colon S' \to T'\rangle$ be two grammars. A *grammar morphism* $\mathcal{F}\colon G \to G'$ is a retyping functor $\mathcal{F}\colon \mathbb{C}\downarrow T \to \mathbb{C}\downarrow T'$ which preserves the structure of G. This means that $\mathcal{F}$ must satisfy $\mathcal{F}(s) = s'$ and for all rules $q = (l \leftarrow\alpha- k -\beta\rightarrow r) \in Q$, the image of q under $\mathcal{F}$ belongs to Q' or is an identity rule, i.e. either $\mathcal{F}(q) \in Q'$ or $\mathcal{F}(\alpha) = \mathcal{F}(\beta) = \mathrm{id}_{\mathcal{F}(k)}$. ☼

Actually, this definition induces a category of grammars (see Corollary E.11 in the appendix). The main advantage of this notion of grammar morphism over the one proposed in [Baldan, 2000] is conceptual clarity. Though there is no objective way to measure this qualitative property, consider the fact that all graph grammar morphisms that occur in the proof of Theorem 6.24 of [Baldan, 2000], which establishes the coreflective characterization of the unfolding for the case of graph grammars, function as if they were retyping functors[66] – although this fact has not been noticed for years. Now, the above definition of grammar morphism makes this fundamental observation about the functorial nature of grammar morphisms explicit and uses structure preserving (retyping) functors as grammar morphisms.

As a "side effect" of this definition of grammar morphisms, the technical restriction to *semi-weighted* grammars of [Baldan, 2000] can be dropped,

[66] See Proposition E.15 on the local determination of occurrence grammars for a property that applies to all grammar morphisms in the proof of Theorem 6.24 of [Baldan, 2000].

and thus, even for the special case of the category of graphs, Theorem 6.6 below improves Theorem 6.24 of [Baldan, 2000]. Moreover, when Petri Nets are considered as an alternative description of rewriting in slice categories of $\mathbb{Set}$, the coreflective characterization of the unfolding of Petri nets can be obtained without the need to go beyond the realm of basic category and without employing notions of enriched category theory and the theory of open map bisimulation [Joyal et al., 1996] as it was done in the recent work [Hayman and Winskel, 2008].

THEOREM 6.6 (Coreflection) Let $G = \langle Q, s\colon S \to T\rangle$ be an $\mathscr{M}$-linear, consuming grammar in some weakly $\mathscr{M}\omega$-adhesive category $\mathbb{C}$ with pushouts along $\mathscr{M}$-morphisms in $\mathrm{Par}(\mathbb{C})$, i.e. for each $m \in \mathscr{M}$ pushouts along $[m\rangle$ exist in $\mathrm{Par}(\mathbb{C})$. Further let $U = \langle Q', s'\colon S \hookleftarrow T'\rangle$ be an unfolding of G with folding morphism $\lambda\colon T' \to T$.

Then for any occurrence grammar $O = \langle Q^\circ, s^\circ\colon S^\circ \to T^\circ\rangle$ and any grammar morphism $\mathcal{F}\colon O \to G$ there is a unique grammar morphism $\mathcal{H}\colon O \to U$ which factors through Σ_λ, i.e. such that $\mathcal{F} = \Sigma_\lambda \circ \mathcal{H}$.

This is the main result of (the first chapter of) this thesis. Its proof is given in Appendix E. It depends on the assumption that each monic match candidate of a rule yields a single pushout rewriting step. In this way, technical complications due to inhibition effects (as in the theory of double pushout rewriting) are avoided.

It is worth mentioning that the proposed notion of grammar morphism as structure preserving retyping functor is central to the proof because it allows to make direct use of the properties of the relevant colimits in weakly $\mathscr{M}\omega$-adhesive categories. The two characteristic properties are universality and stability w.r.t. the embedding into the category of $\mathscr{M}$-partial maps. In fact these two concepts are the main topic of the second chapter on adhesivity and related concepts, as they provide the theoretical foundation for the characterization of the unfolding as a coreflection.

Conclusion and summary of the first part

Taking Mazurkiewicz traces [Mazurkiewicz, 1986] and Petri nets [Petri, 1962] as paradigm models for concurrent computational systems, semantics for the more general single and double pushout rewriting in weakly $\mathscr{M}$-adhesive categories have been presented. The semantics are *concurrent*, i.e. the inherent concurrency of rewriting steps is respected. This is in contrast to interleaving models of computations such as labeled transition systems, which usually contain spurious dependencies between transitions. Of particular interest is the unfolding semantics because it forms the base of widely used analysis and verification techniques [McMillan, 1995, Esparza and Heljanko, 2008]. The first part of this thesis makes it available for a wide range of systems with "graph-like" structure.

The semantic objects are concatenable DPO-processes and occurrence grammars. The former model finite system runs which lead from one state to another one, whereas the latter describe several, possibly incompatible ways in which a system can evolve from a fixed start state. Unfoldings are particular occurrence grammars and they give a complete account of the causality and conflict relations between all events that may occur in a given system.

The two central results about processes are a bijective correspondence between concatenable DPO-processes and switch-equivalence classes (up to isomorphism) and the existence of a suitable composition operation which yields a category of concatenable DPO-processes. These results generalize previous results for graph transformation systems [Baldan et al., 1998a] and Petri nets [Degano et al., 1996]. Preliminary results have been published in the author's [Baldan et al., 2006a].

However, the main result of the first part of this thesis concerns the existence of a procedure which *unfolds* any grammar (which satisfies modest cardinality conditions) into an occurrence grammar which captures all possible evolutions from the start object of the grammar. The central property of the described unfolding procedure is its canonicity in the sense of [Nielsen et al., 1981], i.e. it can be described as a functor which is right adjoint to the inclusion of the category of occurrence grammars into the category of all grammars. This result improves upon previous ones in the area of graph transformation (see e.g. [Baldan et al., 2007]). One corner stone of this result is a suitable notion of grammar morphism (which is a simplified version of the grammar morphisms in the author's [Baldan et al., 2009]).

The validity of the above contributions crucially depends on certain proper-

ties of weakly $\mathscr{M}\omega$-adhesive categories, which often have been simply described as categories with "graph-like" objects. The two main properties of the colimits that exist by definition in any of these categories will be the topic of the second part of this thesis, which provides the category theoretical foundations for the results of the first part.

Categorical Foundation

Partial Van Kampen colimits

7

The first part presented results about the concurrent semantics of adhesive rewriting systems based on the single- and double-pushout approach to rewriting; it also covered some of the relevant proof ideas. However, the assumption was always that the objects that are rewritten congregate into a category with "enough structure", namely a weakly $\mathscr{M}\omega$-adhesive category (such that moreover the associated category of $\mathscr{M}$-partial maps has pushouts along $\mathscr{M}$-morphisms). It remains to establish the necessary results about colimits in these categories.

This section takes a slightly wider perspective and studies a property of colimits which captures one aspect of the "good" behaviour that *all* colimits in the category of graphs exhibit. The relevant fact about colimits in the category of (multi-)graphs (or any topos) is that they are also colimits in the associated category of partial maps, where a partial map is a suitable generalization of a partial function. Colimits with this property will be called *partial Van Kampen* colimits.[67]

Similarly, colimits in the category of simple graphs (or any quasi-topos) remain colimits when embedded into the category of *regular* partial maps, and in the the category of topological spaces the same holds true if the partial maps are restricted to the those with an open domain of definition. Hence, to capture all these cases, the partial Van Kampen colimits are parametric w.r.t. and admissible class of monomorphisms $\mathscr{M}$, where admissibility ensures that the category of $\mathscr{M}$-partial maps is well-defined.

In the context of single- and double-pushout rewriting, pushouts play a fundamental role. Hence, the study of partial Van Kampen colimits starts with pushouts, i.e. partial Van Kampen squares. The main observation is that most results about (proper) Van Kampen squares[68] which have been established in [Lack and Sobociński, 2004] already hold for the weaker *partial* ones. In particular, (suitable versions of) sequential (and parallel) commutativity follow already from the properties of partial Van Kampen squares.

The central definition of this section introduces *$\mathscr{M}$-partial Van Kampen* colimits in a category $\mathbb{C}$ with an admissible class of monomorphisms $\mathscr{M}$

[67] This terminology can motivated in the context of [Heindel and Sobociński, 2009] as follows: while (proper) Van Kampen colimits are exactly those colimits which are "preserved" by the embedding in the bicategory of spans, the partial ones are those which "induce" colimits in the properly smaller sub-bicategory of partial maps.

[68] See Definition 2.20 and Figure 25.

(see Definition 2.48). They are exactly those colimits which are preserved by the graphing functor into the category of $\mathscr{M}$-partial maps. An axiomatic, elementary characterization of $\mathscr{M}$-partial Van Kampen colimits is given in Theorem B.4, which establishes the connection to the elementary Van Kampen square characterization of [Lack and Sobociński, 2005] (see also Figures 25 and 61).

Overview of the section. The main goal of this section is to propose alternatives for the relatively rare Van Kampen squares [Lack and Sobociński, 2005] (and colimits [Heindel and Sobociński, 2009]). The alternatives must of course enjoy all properties that are necessary for the results of the first part of this thesis. A first candidate are the partial Van Kampen colimits which will be defined in Section 7.1.

Section 7.2 studies the special case of pushouts, i.e. partial Van Kampen squares. The first result is Proposition 7.3, which gives an alternative characterization of partial Van Kampen squares that is much closer to the elementary definition of Van Kampen squares given in [Lack and Sobociński, 2004]. Further, an analysis of the defining property of (partial) Van Kampen squares is given: it can be split into ($\mathscr{M}$-)universality (which is also known as pullback stability along morphisms in $\mathscr{M}$) and a second complementary property. A more comprehensive comparison with (other variations of) adhesive categories in the literature is deferred to Section 8.

Next, Section 7.3 lists evidence that partial Van Kampen squares are a viable alternative to (proper) Van Kampen squares. This is done by establishing the major part of the properties of Van Kampen squares and pushouts in adhesive categories that have been established in [Lack and Sobociński, 2005] for the more general partial Van Kampen pushouts.

Finally, Section 7.4 studies partial Van Kampen pushouts which are also stable under pullback along *arbitrary* morphisms (and not only along morphisms of the class $\mathscr{M}$). If one adds this property (as it has been done in the definition of weakly adhesive categories) one obtains certain distributivity laws in subobject lattices. This restricted distributivity actually suffices for the results of the first part of the thesis. In other words, though "full" distributivity of subobject lattices as known from adhesive categories is a very convenient property, it seems that it is not essential.

7.1 The definition

The definition of partial Van Kampen colimits is based on the graphing functor. Recall that in a category $\mathbb{C}$ with an admissible class of monomorphisms $\mathscr{M}$, the graphing functor $\Gamma\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C}, \mathscr{M})$ is the identity on objects, and it maps each $\mathbb{C}$-morphism $f\colon C \to D$ to its functional relation $\Gamma(f)\colon C \rightharpoonup D$. With the notion of the graphing functor at hand, the details of the general definition of $\mathscr{M}$-partial Van Kampen colimit are as follows.

☼ Definition 7.1 ($\mathscr{M}$-partial Van Kampen / $\mathscr{M}$-hereditary colimits) Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$; let $\mathrm{Par}(\mathbb{C}, \mathscr{M})$ be the associated category of $\mathscr{M}$-partial maps; further let I be a graph, let $\mathcal{D}\colon I \to \mathbb{C}$ be a diagram, and let $\mathbb{c}\colon \mathcal{D} \to \Delta_D$ be its colimit.

Then the colimit $\mathbb{c}$ is *$\mathscr{M}$-partial Van Kampen* (PVK) or *$\mathscr{M}$-hereditary* if it is preserved by the graphing functor $\Gamma_{\mathscr{M}}\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C}, \mathscr{M})$.[69] The category $\mathbb{C}$ *has $\mathscr{M}$-partial Van Kampen colimits (of shape I)* if it has colimits (of shape I) and all these colimits are $\mathscr{M}$-partial Van Kampen. ☼

Partial Van Kampen squares are are also known as hereditary pushouts, as discussed in [Kennaway, 1990]; for the special case of pushouts, the latter work gives a characterization which is similar to the one in Example 7.2 below (cf. [Kennaway, 1990, Definition 3.3]).[70]

7.2 Partial Van Kampen squares

As a first example of partial Van Kampen colimits, consider partial Van Kampen squares, i.e. PVK-pushouts; a "direct", elementary characterization of these pushouts given in Example 7.2 without reference to the graphing functor. Partial Van Kampen pushouts can be seen as a combination of ideas from [Lack and Sobociński, 2005], which introduces (proper) Van Kampen squares, and [Ehrig and Prange, 2006], which generalizes the latter to "weak" Van Kampen squares. However, as mentioned above, pushouts in partial map categories have already been discussed in [Kennaway, 1990].

☙ Example 7.2 (Partial Van Kampen squares) Let $\mathbb{C}$ be a category with pullbacks along monomorphisms, let $B \leftarrow f - A - m \rightarrow C$ be a $\mathbb{C}$-span and let

[69] That $\Gamma_{\mathscr{M}}$ preserves the colimit $\mathbb{c}$ means that $\Gamma_{\mathscr{M}}\mathbb{c} = \{\Gamma_{\mathscr{M}}(\mathbb{c}_i)\colon \mathcal{D}_I(i) \rightharpoonup D\}_{i \in V_I}$ is a colimit of the diagram $\Gamma_{\mathscr{M}} \circ \mathcal{D}$ in $\mathrm{Par}(\mathbb{C}, \mathscr{M})$.

[70] However the proofs of [Kennaway, 1990] seem to be based on some implicit assumptions about subobject lattices since the phrase *the largest possible subobject* is used at several places in the proofs.

$B \mathrel{-n\to} D \mathrel{\leftarrow g-} C$ be its pushout, which results in the pushout square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$. In

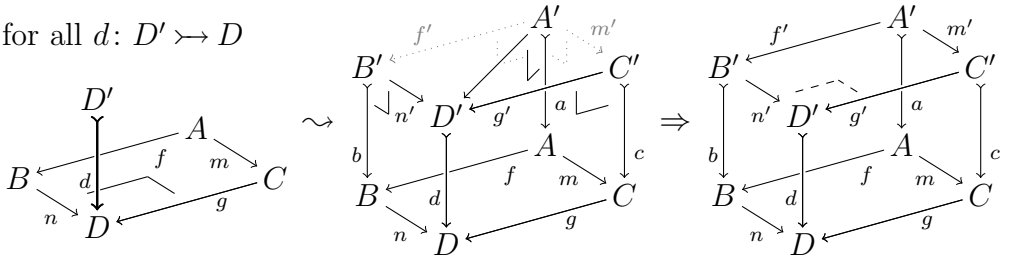


Figure 59: Pullback stability of pushouts along monomorphisms

this situation, Figure 59 illustrates what it means that this pushout is stable under pullback along any $\mathbb{C}$-monomorphism $d\colon D' \rightarrowtail D$.

The middle diagram in Figure 59 shows how the bottom square is pulled back along d. First, take the pullbacks of $d\colon D' \rightarrowtail D$ along $n\colon B \to D$, $g\colon C \to D$, and the diagonal $g \circ m\colon A \to D$; this results in pullback objects B', C', and A', respectively. Second, pullback splitting yields mediating morphisms $m'\colon A' \to C'$ and $f'\colon A' \to B'$. The result is a new square ${}^{B'\leftarrow A'\searrow}_{\searrow D'\leftarrow C'}$ on top of a cube the four lateral sides of which are pullbacks. Finally, that the bottom pushout is stable under pullback means that the top square ${}^{B'\leftarrow A'\searrow}_{\searrow D'\leftarrow C'}$, which arose by pulling back the bottom pushout square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$ along $d\colon D' \rightarrowtail D$, is not only a commuting square but actually a pushout square. It is proved below in Lemma B.3 that partial Van Kampen pushouts are stable under pullback along monomorphisms.

In the opposite direction, start with the same pushout square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$, a monomorphisms $a\colon A' \rightarrowtail A$, and two pullback complements $C \mathrel{\leftarrow c\prec} C' \mathrel{\leftarrow m'-} A'$ and $B \mathrel{\leftarrow b\prec} B' \mathrel{\leftarrow f'-} A'$ of $C \mathrel{\leftarrow m-} A \mathrel{\leftarrow a\prec} A'$ and $B \mathrel{\leftarrow f-} A \mathrel{\leftarrow a\prec} A'$, respectively, with the additional requirement that $b\colon B' \rightarrowtail B$ and $c\colon C' \rightarrowtail C$ are monic as well (see the left hand diagram in Figure 60). In this situation, constructing a pushout $B' \mathrel{-n'\to} D' \mathrel{\leftarrow g'-} C'$ of $B' \mathrel{\leftarrow f'-} A' \mathrel{-m'\to} C'$ yields the pushout square ${}^{B'\leftarrow A'\searrow}_{\searrow D'\leftarrow C'}$ and moreover a mediating morphism $d\colon D' \to D$ (see the middle diagram in Figure 60). If the bottom pushout $B \mathrel{-n\to} D \mathrel{\leftarrow g-} C$ is a PVK-pushout, then the morphism d is monic and the front squares are pullback squares, as illustrated in the right hand diagram in Figure 60; this is a consequence of Lemma B.2 below.

Finally, as formulated in Proposition 7.3, the two properties described above and illustrated in Figures 59 and 60 actually imply that the pushout is partial Van Kampen. Summarizing, this means that the pushout square

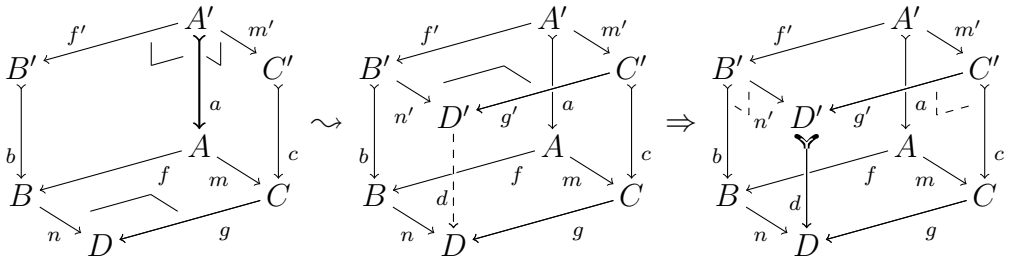


Figure 60: The "opposite" of pullback stability: merging pullback complements

${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$ based on the span $B \leftarrow f- A -m\rightarrow C$ and its pushout $B -n\rightarrow D \leftarrow g- C$ is a partial Van Kampen square if and only if it satisfies the property illustrated in Figure 61. This means that for each commutative cube on top of the pushout

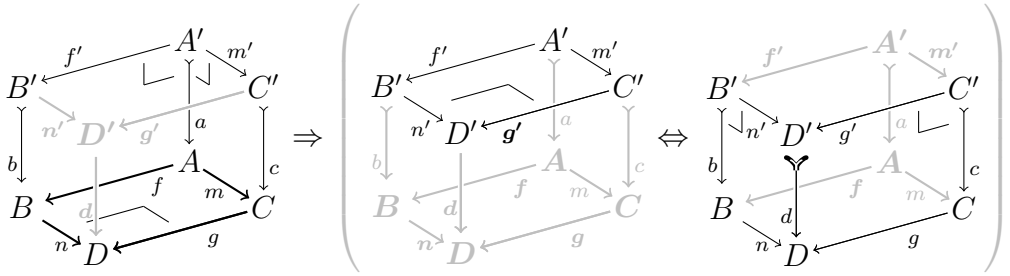


Figure 61: Partial Van Kampen square property

square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$ as shown in Figure 61 on the left, with pullback squares as back faces and all three arrows a, b and c monic, its top face is a pushout square if and only if the front faces are pullback squares and the morphism d is monic.

The crucial observation is that in the left cube diagram of Figure 61, the partial maps $[c,g'\rangle\colon C \rightharpoonup D'$, $[a,g'\circ m'\rangle\colon A \rightharpoonup D'$ and $[b,n'\rangle\colon B \rightharpoonup D'$ form a cocone of the span $B \leftarrow \Gamma f- A -\Gamma m\rightarrow C$ in $\mathrm{Par}(\mathbb{C})$. The details are described in the proof of Lemma B.2. Indeed, Example 7.2 also holds if one only considers monomorphisms of an admissible class $\mathscr{M}$.

Proposition 7.3 ($\mathscr{M}$-partial Van Kampen square characterization) Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$; further assume that $\mathbb{C}$ has pushouts.

Then $\mathbb{C}$-pushouts are $\mathscr{M}$-partial Van Kampen pushouts if and only if for each span $B \leftarrow f- A -g\rightarrow C$, pushout $B -h\rightarrow D \leftarrow k- C$, and commutative cube over the pushout square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$ with pullback squares ${}^{B'}_{B}\downarrow\!\!\leftarrow\!\!\downarrow{}^{A'}_{A}$ and ${}^{A'}_{A}\downarrow\!\!\rightarrow\!\!\downarrow{}^{C'}_{C}$ as

back faces where $b\colon B' \hookrightarrow B$ and $c\colon C' \hookrightarrow C$ are $\mathscr{M}$-morphisms as depicted on the left in Figure 61, it is the case that the top face $\begin{smallmatrix} B' \leftarrow A' \searrow \\ \searrow D' \leftarrow C' \end{smallmatrix}$ is a pushout square if and only if the front faces $\begin{smallmatrix} B' \\ \downarrow \\ B \end{smallmatrix} \square \begin{smallmatrix} \rightarrow \\ \downarrow \\ \rightarrow \end{smallmatrix} \begin{smallmatrix} D' \\ \downarrow \\ D \end{smallmatrix}$ and $\begin{smallmatrix} D' \\ \downarrow \\ D \end{smallmatrix} \square \begin{smallmatrix} \leftarrow \\ \downarrow \\ \leftarrow \end{smallmatrix} \begin{smallmatrix} C' \\ \downarrow \\ C \end{smallmatrix}$ are pullback squares and the morphism $d\colon D' \hookrightarrow D$ is a monomorphisms of the class $\mathscr{M}$.

Proof. This proposition is a special case of Theorem B.4 below. □

A pushout square which satisfies the property in the consequence of this proposition will also be referred to as an $\mathscr{M}$-partial Van Kampen square. This characterization of PVK pushouts has the advantage that the category $\mathbb{C}$ is not required to have all pushouts. However, for the purpose of rewriting via single- or double-pushout grammars, this difference is marginal. The reason is that in such a setting the category in which rewriting takes place is usually assumed to have "enough" pushouts, e.g. all pushouts along monomorphisms or $\mathscr{M}$-morphisms exist.

In the following study of categories with partial Van Kampen pushouts, the latter will be called hereditary pushout categories, for the sake of a shorter name.

☼ Definition 7.4 ($\mathscr{M}$-hereditary pushout categories) Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$. Then the category $\mathbb{C}$ is an *$\mathscr{M}$-hereditary pushout category* or *$\mathscr{M}$-HEPO category* if

- the category $\mathbb{C}$ has pushouts,
- all pushouts yield partial Van Kampen squares, i.e. the graphing functor $\Gamma_{\mathscr{M}}\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C}, \mathscr{M})$ preserves pushouts.

☼

The main property of pushouts in HEPO categories can be split up into two properties, namely *$\mathscr{M}$-universality* (see Definition 7.5) and the *pullback merging* (see Definition 7.7).

☼ Definition 7.5 ($\mathscr{A}$-universality of pushouts) Let $\mathbb{C}$ be any category and let $\mathscr{A} \subseteq \mathrm{ar}(\mathbb{C})$ be any class of $\mathbb{C}$-arrows. Further let $B \leftarrow f - A - m \rightarrow C$ be a $\mathbb{C}$-span, and let $B - n \rightarrow D \leftarrow g - C$ be a pushout of $B \leftarrow f - A - m \rightarrow C$.

Then the pushout $B - n \rightarrow D \leftarrow g - C$ is *$\mathscr{A}$-universal* or *stable under pullback along $\mathscr{A}$-morphisms* if for any commutative cube over the pushout square $\begin{smallmatrix} B \leftarrow A \searrow \\ \searrow D \leftarrow C \end{smallmatrix}$ having four pullback squares as lateral faces as shown in the middle diagram of Figure 62, the top face $\begin{smallmatrix} B' \leftarrow A' \searrow \\ \searrow D' \leftarrow C' \end{smallmatrix}$ is a pushout square provided that $d\colon D' \to D$ belongs to $\mathscr{A}$. ☼

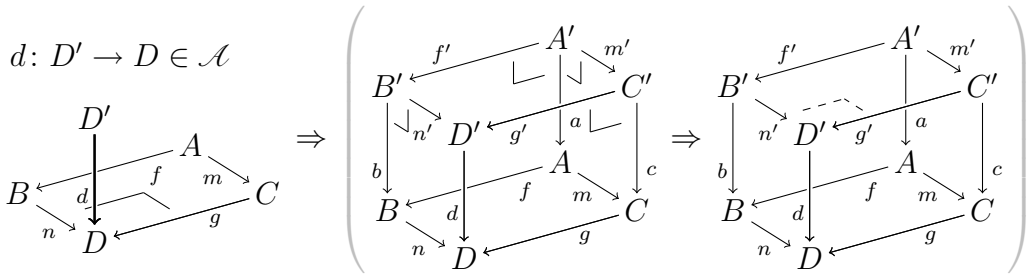


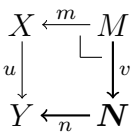
Figure 62: Universality of pushouts where $d\colon D' \to D$ is an $\mathscr{A}$-morphism

As a direct consequence, a pushout is universal in a category $\mathbb{C}$ if it is $\mathrm{ar}(\mathbb{C})$-universal (cf. 9.3). Universality of colimits is a well-known phenomenon, since for example all colimits in cartesian, locally cartesian closed categories are universal, which can be shown using the proof of Corollary 9.5. Further all (proper) Van Kampen squares are trivially universal pushout squares. The idea of considering any class of morphisms is also present in [Monserrat et al., 1997]: condition (S3b) in the latter work exactly corresponds to universality of pushouts.

The "opposite" of universality, which features prominently in the theory descent as presented in [Janelidze et al., 2004], will recur in Lemma B.2. For the special case of pushouts, it will be called the *pullback merging property*, which is formulated in terms of $\mathscr{A}$-pullback complements.

☼ Definition 7.6 (Pullback complements) Let $\mathbb{C}$ be any category and let $u\colon X \to Y$ and $m\colon M \to X$ be $\mathbb{C}$-morphisms, which thus form a composable pair $Y \leftarrow_u - X \leftarrow_m - M$.

An *(arbitrary*[71]*) pullback complement* for $Y \leftarrow_u - X \leftarrow_m - M$ is another composable pair $Y \leftarrow_n - N \leftarrow_v - M$ such that the pair $X \leftarrow_m - M \rightarrow_v N$ is a pullback of $X \rightarrow_u Y \leftarrow_n - N$; thus, such a pullback complement gives rise to a pullback square $\begin{smallmatrix} X & \leftarrow & M \\ \downarrow & \lrcorner & \downarrow \\ Y & \leftarrow & N \end{smallmatrix}$.



An *$\mathscr{A}$-pullback complement* for the composable pair $Y \leftarrow_u - X \leftarrow_m - M$ w.r.t. to a class $\mathscr{A} \subseteq \mathrm{ar}(\mathbb{C})$ of $\mathbb{C}$-arrows is a pullback complement $Y \leftarrow_n - N \leftarrow_v - M$ for $Y \leftarrow_u - X \leftarrow_m - M$ such that $n\colon N \to Y$ is an $\mathscr{A}$-morphism.[72] ☼

[71] The qualification *arbitrary* is only necessary to distinguish arbitrary pullback complements from the pullback complements as defined in [Dyckhoff and Tholen, 1987]. In this thesis, what is called *pullback complement* in the latter work is referred to as *final pullback complement* in this thesis.

[72] One might want to require the class $\mathscr{A}$ to be stable under pullback.

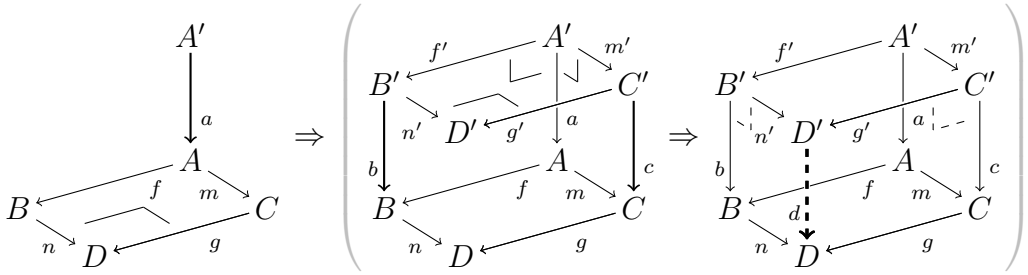


Figure 63: Pullback merging property

☼ Definition 7.7 (Pullback merging property) Let $\mathbb{C}$ be any category and let $\mathscr{A} \subseteq \mathrm{ar}(\mathbb{C})$ be any class of $\mathbb{C}$-arrows. Further let $B \leftarrow_f - A -_m \rightarrow C$ be a $\mathbb{C}$-span, and let $B -_n \rightarrow D \leftarrow_g - C$ be a pushout of $B \leftarrow_f - A -_m \rightarrow C$.

Then the pushout *merges $\mathscr{A}$-pullback complements* or satisfies the *$\mathscr{A}$-pullback merging property* if for any commutative cuboid diagram over the pushout square ${}^{B \leftarrow A \searrow}_{\searrow D \leftarrow C}$ which has pullback squares as back faces and a pushout square ${}^{B' \leftarrow A' \searrow}_{\searrow D' \leftarrow C'}$ as top face as shown in Figure 63 with the additional requirement that the two morphisms $b\colon B' \to B$ and $c\colon C' \to C$ are $\mathscr{A}$-morphisms, it is the case that these conditions imply that the mediating morphism $d\colon D' \to D$ is an $\mathscr{A}$-morphism and the front faces are pullbacks. ☼

The idea that pushouts merge certain pullback complements is related to condition (S3a) of [Monserrat et al., 1997]. In the latter work the discussion is not specialized to partial map categories but also other suitable subcategories of the span-category are considered; however, as the authors of the latter work write [Monserrat et al., 1997, page 3],

> we do not need to impose that $\mathscr{M}$ be a class of monomorphisms (although, to be honest, we do not know any natural example [...] with $\mathscr{M}$ containing non-monomorphic morphisms), although when $\mathscr{M}$ is a class of monomorphisms then some results can be strengthened.

7.3 Properties of partial Van Kampen squares

The major part of the properties of (proper) Van Kampen squares that have been established in [Lack and Sobociński, 2005] already hold for partial Van Kampen squares; moreover the proofs of [Lack and Sobociński, 2005] can be adapted without major changes. In other words, the greater generality of

partial Van Kampen squares does not lead to theoretical complications. In particular the basic theorems of the concurrency theory of DPO rewriting, namely sequential (and parallel) commutativity, hold in $\mathscr{M}$-HEPO categories.

To facilitate a detailed comparison with adhesive and (weak) adhesive HLR-categories, a large part of the basic properties of Van Kampen squares and adhesive categories presented in [Lack and Sobociński, 2005] will be established for PVK squares and HEPO categories, respectively. The properties are divided into *basic properties* and HLR-*properties*.

7.3.1 Basic properties

Before discussing those properties of partial Van Kampen squares that are rather specific to the concurrency theory of double-pushout rewriting,[73] the major part of the basic facts about (proper) Van Kampen squares presented in [Lack and Sobociński, 2005] will be established for $\mathscr{M}$-partial Van Kampen squares. For the remainder of this section, fix a category $\mathbb{C}$ with an admissible class of monomorphisms $\mathscr{M}$.

The first basic property is the so-called Pushout Pullback Decomposition property. In fact, the next lemma subsumes Lemmas 4.6 and 4.7 of [Lack and Sobociński, 2005]; it alone suffices to proof the parallel and sequential commutativity (see e.g. [Habel et al., 2001]), in hereditary pushout categories.[74]

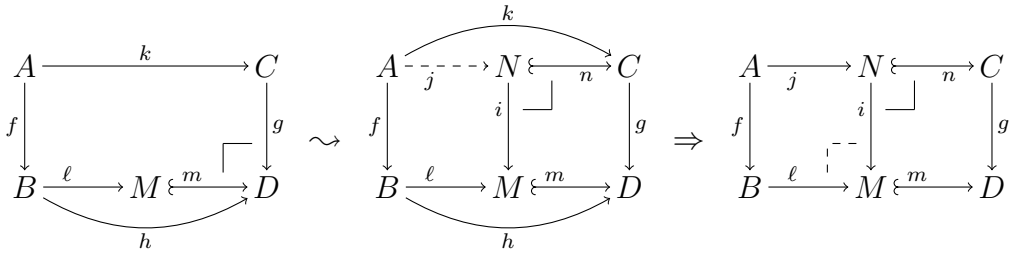


Figure 64: Decomposing a PVK-pushout by pullback along an $\mathscr{M}$-morphism

LEMMA 7.8 (Pushout pullback decomoposition) Let $B \leftarrow f - A - k \rightarrow C$ be a cospan, let $B - h \rightarrow D \leftarrow g - C$ be an $\mathscr{M}$-universal pushout of $B \leftarrow f - A - k \rightarrow C$,

[73] More explicitly, this sentence refers the family of HLR-systems surveyed in [Padberg, 1993].

[74] This lemma also is related to the more specific $\mathscr{M}$-pushout-pullback decomposition property, which is required for HLR1 and HLRE (see [Padberg, 1993]).

and let $h\colon B \to D$ factor as $m \circ \ell$ for morphisms $l\colon B \to M$ and $m\colon M \rightarrowtail D$ such that m is an $\mathscr{M}$-morphism (see the left diagram in Figure 64). Further let $M \leftarrow i - N \rightarrowtail n \rightarrow C$ be the pullback of $M \rightarrowtail m \rightarrow D \leftarrow g - C$, and let $j\colon A \to N$ be the unique morphism satisfying $i \circ j = \ell \circ f$ and $n \circ j = k$ as shown in the middle diagram in Figure 64.

Then the cospan $B - \ell \rightarrow M \leftarrow i - N$ is a pushout of $B \leftarrow f - A - j \rightarrow N$.

Proof. The proof of this lemma is essentially the same as the one given for Lemmas 4.6 and 4.7 in [Lack and Sobociński, 2005]. □

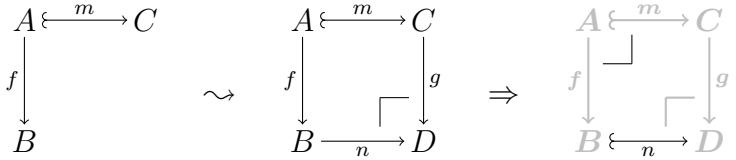


Figure 65: The Pushout-Pullback Lemma

Another property that (proper) Van Kampen squares share with partial ones, is the Pushout-Pullback Lemma (cf. [Ehrig and Kreowski, 1979]), which can be proven in analogy to Lemma 2.3 of [Lack and Sobociński, 2005]. Here a (more verbose version of the) proof is given as it is one of the few proofs that directly depend on the pullback complement merging property of (partial) Van Kampen squares.

Lemma 7.9 (Pushout-Pullback Lemma) Let $B \leftarrow f - A \rightarrowtail m \rightarrow C$ be a $\mathbb{C}$-span such that m is an $\mathscr{M}$-morphism; let $B - n \rightarrow D \leftarrow g - C$ be a PVK pushout of $B \leftarrow f - A \rightarrowtail m \rightarrow C$.

Then the morphism n belongs to the class $\mathscr{M}$ and $B \leftarrow f - A \rightarrowtail m \rightarrow C$ is a pullback of $B \rightarrowtail n \rightarrow D \leftarrow g - C$ (see Figure 65).

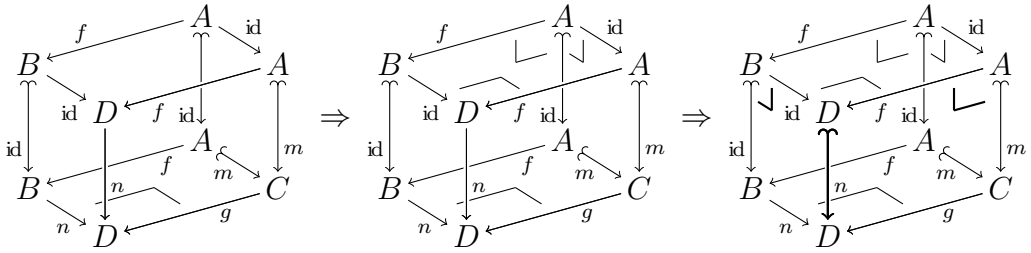


Figure 66: Illustration of the proof of Lemma 7.9

Proof. Take the pushout square ${}^{B \leftarrow A \searrow}_{\searrow D \leftarrow C}$ as the base of the cube diagram shown on the left in Figure 66. In this cube diagram, $A \leftarrow \mathrm{id} - A - \mathrm{id} \rightarrow A$ is a pullback of $C \leftarrow m \rightarrowtail A \hookleftarrow m \rightarrow C$ since m is monic; further $A \leftarrow \mathrm{id} - A - f \rightarrow B$ is a pullback of $B \leftarrow f - A - \mathrm{id} \rightarrow A$, and finally $B \leftarrow f - A - \mathrm{id} \rightarrow A$ is the pushout of $A \leftarrow \mathrm{id} - A - f \rightarrow B$ (see the middle diagram of Figure 66).

The bottom pushout square ${}^{B \leftarrow A \searrow}_{\searrow D \leftarrow C}$ is an $\mathscr{M}$-partial Van Kampen square by assumption, and hence Proposition 7.3 is applicable; this implies that the front faces of the cube are pullbacks as shown on the right in Figure 66. This in turn means that n is an $\mathscr{M}$-morphism and the span $B \leftarrow f - A \hookleftarrow m \rightarrow C$ is a pullback of $B \hookleftarrow n \rightarrow D \leftarrow g - C$. □

Pullbacks that arise from PVK pushouts along $\mathscr{M}$-morphisms have the following feature: given a span $B \leftarrow f - A \hookleftarrow m \rightarrow C$ with PVK pushout $B \hookleftarrow n \rightarrow D \leftarrow g - C$ such that $m \in \mathscr{M}$, the composable pair $A - f \rightarrow B \hookleftarrow n \rightarrow D$ can be "recovered" from $A \hookleftarrow m \rightarrow C - g \rightarrow D$ (up to isomorphism); the reason is that the pushout complement is actually the *greatest pullback complement*[75] of the composable pair $A \hookleftarrow m \rightarrow C - g \rightarrow D$. This fact also ensures that pushout complements of a composable pair such as $A \hookleftarrow m \rightarrow C - g \rightarrow D$ are unique up to isomorphism. The details are as follows.

☼ Definition 7.10 (Greatest pullback complements)
Let $u \colon X \to Y$ be a morphism and let $m \colon M \hookrightarrow X$ be an $\mathscr{M}$-morphism.

A *greatest $\mathscr{M}$-pullback complement* (GPBC) for the pair $Y \leftarrow u - X \leftarrow m \rightarrowtail M$ is an $\mathscr{M}$-pullback complement $Y \leftarrow n \rightarrowtail N \leftarrow v - M$ such that for any other $\mathscr{M}$-pullback complement $Y \leftarrow \ell \rightarrowtail L \leftarrow w - M$, the inclusion $n \sqsupseteq \ell$ holds in $\mathrm{Sub}_{\mathscr{M}}(Y)$. ☼

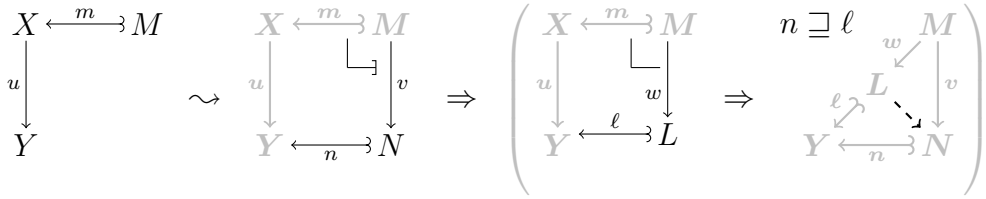


Figure 67: Greatest pullback complements

In other words, the Pushout-Pullback Lemma can be strengthened as follows.

[75] One could also use the slightly more complicated mono-final pullback complements of Definition 2.41 – in fact this is what is done in [Lack and Sobociński, 2004].)

Lemma 7.11 (Greatest pullback complements from hereditary pushouts) Let $B \leftarrow f - A \hookleftarrow m \rightarrow C$ be a $\mathbb{C}$-span such that $m \in \mathscr{M}$, and let $B \hookleftarrow n \rightarrow D \leftarrow g - C$ be a PVK pushout of $B \leftarrow f - A \hookleftarrow m \rightarrow C$.

Then $D \leftarrow n \twoheadrightarrow B \leftarrow f - A$ is a greatest $\mathscr{M}$-pullback complement of the composable pair $D \leftarrow g - C \leftarrow m \twoheadrightarrow A$.

Proof. The proof of this lemma is essentially the same as the one given for Lemma 2.8 of [Lack and Sobociński, 2005]. □

Corollary 7.12 (Uniqueness of pushout complements) Let $\mathbb{C}$ be a hereditary pushout category. Let $D \leftarrow g - C \leftarrow m \twoheadrightarrow A$ be a composable pair of morphisms with $m \in \mathscr{M}$.

Then pushout complements of $D \leftarrow g - C \leftarrow m \twoheadrightarrow A$ are unique up to isomorphism, i.e. given any two pushout complements $D \leftarrow n \twoheadrightarrow B \leftarrow f - A$ and $D \leftarrow n' \twoheadrightarrow B' \leftarrow f' - A$ of $D \leftarrow g - C \leftarrow m \prec A$, there exists an isomorphism $i \colon B \to B'$ such that both $n' = n \circ i$ and $f = i \circ f'$ hold.

Proof. By Lemma 7.11, both $D \leftarrow n \twoheadrightarrow B \leftarrow f - A$ and $D \leftarrow n' \twoheadrightarrow B' \leftarrow f' - A$ are greatest pullback complements of $D \leftarrow g - C \leftarrow m \twoheadrightarrow A$ and hence both $n \sqsubseteq n'$ and $n' \sqsubseteq n$ hold, i.e. $n \equiv n'$ in $\mathrm{Sub}(Y)$. □

Concluding the list of basic properties, all $\mathscr{M}$-morphisms can be shown to be regular; the proof is again adapted from [Lack and Sobociński, 2005].

Lemma 7.13 (Regular monomorphisms) In any $\mathscr{M}$-hereditary pushout category, all $\mathscr{M}$-morphisms are regular.

Proof. Let $m \colon M \hookrightarrow A$ in $\mathbb{C}$ be an $\mathscr{M}$-morphisms; let $A - p \rightarrow D \leftarrow q - A$ be the pushout of $A \leftarrow m \twoheadrightarrow M \hookleftarrow m \rightarrow A$. Then, by Lemma 7.9, the span $A \leftarrow m \twoheadrightarrow M \hookleftarrow m \rightarrow A$ is a pullback of $A - p \rightarrow D \leftarrow q - A$, which means that $m \colon M \hookrightarrow A$ is the equalizer of $p, q \colon A \rightrightarrows D$. □

Corollary 7.14 Hereditary $\mathscr{M}\!\mathit{ono}$-pushout categories are balanced[76].

Proof. This is an immediate consequence of Lemma 7.13 and the general fact that (in any category) every epic regular monomorphism is an isomorphism. □

[76] A category is *balanced*, if each bimorphism is an isomorphism where a *bimorphism* is a morphism that is both monic and epic.

7.3.2 High-level-replacement properties

The list of characteristic properties related to adhesivity in the context of single and double pushout rewriting includes the following three items: the so-called Twisted Triple Pushout Condition, the Cube Pushout Pullback Lemma and the the so-called special pullback-pushout property [Ehrig and Kreowski, 1979]. The third property can be proofed by reusing the proof of Corollary 8.6 below; here the first two items are established for $\mathscr{M}$-HEPO categories. All three facts are best understood in the context of high-level-replacement systems [Ehrig and Löwe, 1993].[77]

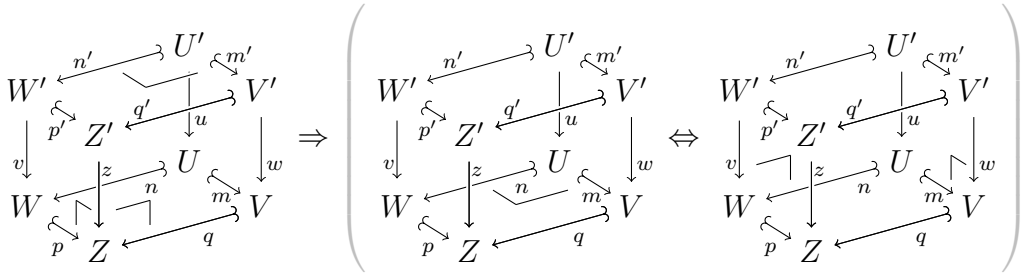


Figure 68: Cube Pushout Pullback Lemma

The Cube Pushout Pullback Lemma is illustrated in Figure 68. It is a direct consequence of characterization of partial Van Kampen squares given in Example 7.2.

LEMMA 7.15 (Cube Pushout Pullback Lemma) For any cube diagram in which all but the vertical morphisms are $\mathscr{M}$-morphisms as shown Figure 68 such that the font faces are two partial Van Kampen squares and the top face is a pullback, the following is true: the bottom face is a pullback if and only if the back faces are pushouts.

Proof. This lemma directly follows from the partial Van Kampen property as described in the proof of Lemma 8.4 of [Lack and Sobociński, 2005]. □

Next, the twisted triple pushout property holds if the category $\mathbb{C}$ has $\mathscr{M}$-partial Van Kampen pushouts.[78] It is established by the following lemma, which can be described roughly as giving sufficient conditions under which

[77] A list of theorems of double pushout transformation together with the properties that are used in their proofs can be found in Table 3.1 of [Padberg, 1993].

[78] Actually, in the situation of Figure 69, it is enough if the two pushouts $E \xrightarrow{k} F \xleftarrow{q \circ g} C$ and $C \xrightarrow{g} D \xleftarrow{v} Z$ are partial Van Kampen.

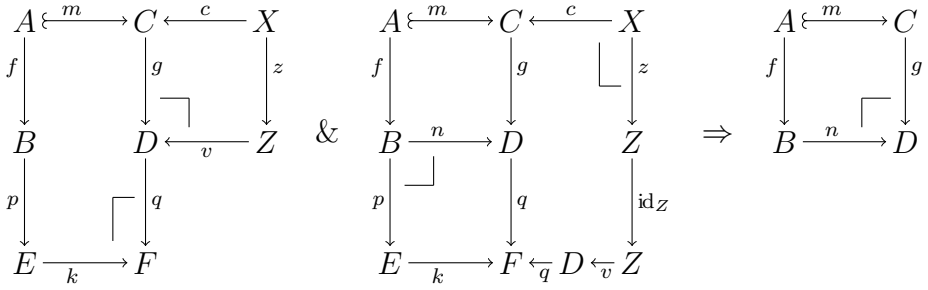


Figure 69: Twisted-Triple-Pushout property

the morphism $q\colon D \to F$ can be used to split the left hand pushout ${}^{A}_{E}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{C}_{F}$ in Figure 69.

Lemma 7.16 (Twisted-Triple-Pushout Lemma) Let $A \mathrel{-}f\!\!\rightarrow B \mathrel{-}p\!\!\rightarrow E$ be a pair of morphisms, let $A \mathrel{\hookleftarrow}\!m\!\!\rightarrow C$ be an $\mathscr{M}$-morphism, let $C \mathrel{-}g\!\!\rightarrow D \mathrel{-}q\!\!\rightarrow F$ be a pair of morphisms, and let $E \mathrel{-}k\!\!\rightarrow F$ be a morphism such that $E \mathrel{-}k\!\!\rightarrow F \leftarrow\!q\circ g\!- C$ is a PVK pushout of $E \leftarrow\!p\circ f\!- A \mathrel{\hookleftarrow}\!m\!\!\rightarrow C$, thus giving rise to the pushout square ${}^{A}_{E}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{C}_{F}$. Further let $C \leftarrow\!c\!- X \mathrel{-}z\!\!\rightarrow Z$ be a span and let $D \leftarrow\!v\!- Z$ be an arrow such that $C \mathrel{-}g\!\!\rightarrow D \leftarrow\!v\!- Z$ is an $\mathscr{M}$-PVK pushout of $C \leftarrow\!c\!- X \mathrel{-}z\!\!\rightarrow Z$ (this results in another pushout square ${}^{C}_{D}\!\downarrow\!{}^{\leftarrow}_{\leftarrow}\!\downarrow{}^{X}_{Z}$) and also $C \leftarrow\!c\!- X \mathrel{-}z\!\!\rightarrow Z$ is a pullback of $C \mathrel{-}q\circ g\!\!\rightarrow F \leftarrow\!q\circ v\!- Z$, which gives rise to the pullback square ${}^{C}_{F}\!\downarrow\!{}^{\leftarrow}_{\leftarrow}\!\downarrow{}^{X}_{Z}$. Finally let $B \mathrel{-}n\!\!\rightarrow D$ be an arrow such that the equation $g \circ m = n \circ f$ holds and the span $E \leftarrow\!p\!- B \mathrel{-}n\!\!\rightarrow D$ is a pullback of $E \mathrel{-}k\!\!\rightarrow F \leftarrow\!q\!- D$, thus yielding another pullback square ${}^{B}_{E}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{D}_{F}$.

Then $B \mathrel{-}n\!\!\rightarrow D \leftarrow\!g\!- C$ is a pushout of $B \leftarrow\!f\!- A \mathrel{\hookleftarrow}\!m\!\!\rightarrow C$, which results in the pushout square ${}^{A}_{B}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{C}_{D}$.

Proof. The proof of this lemma is essentially the same as the one given for Lemma 8.5 of [Lack and Sobociński, 2005]. □

This concludes the list of the relevant properties of $\mathscr{M}$-partial Van Kampen pushouts. One may actually wonder why one needs more than these properties. The main reason is that it is convenient to have also "well-behaved" subobject posets, as discussed next.

7.4 Adding universality

A well-known fact about the category of sets is that every power set poset actually forms a Boolean algebra. Further, in any topos, subobject posets still

form distributive lattices. The latter fact has been established for adhesive categories in [Lack and Sobociński, 2005]; its proof depends on universality of pushouts along monomorphisms. In fact, distributivity can also be obtained in $\mathscr{M}$-HEPO categories (at least for joins of $\mathscr{M}$-subobjects) provided that pushouts are not only partial Van Kampen but are also universal.

☼ DEFINITION 7.17 (Universal, hereditary pushout categories) Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$. Then $\mathbb{C}$ is a *universal, hereditary pushout category w.r.t.* $\mathscr{M}$ ($\mathscr{M}$-UHEPO) if it is a hereditary pushout category w.r.t. $\mathscr{M}$ such that additionally all pushouts are universal. ☼

In such categories, joins cannot be computed effectively in general as shown in Example 2.47. Nevertheless, one obtains a weakened (but sufficiently strong) version of $\mathscr{M}$-effective unions and distributivity as made precise by the following Proposition.

◌ PROPOSITION 7.18 (Pseudo-effective unions and distributivity) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $Z \in \mathbb{C}$ be an object, and let $[a], [b] \in \mathrm{Sub}_{\mathscr{M}}(Z)$ be two $\mathscr{M}$-subobjects.

Then the join $[a] \sqcup [b]$ of $[a]$ and $[b]$ exists at least in the poset $\mathrm{Sub}(Z)$ of arbitrary subobjects, and can be computed as the pushout over the intersection. Moreover, the equation

$$c \sqcap (a \sqcup b) = (c \sqcap a) \sqcup (c \sqcap b)$$

holds for any arbitrary subobject $[c] \in \mathrm{Sub}(Z)$.

Proof. This proposition can be established by adapting the proofs of Theorem 5.1 and Corollary 5.2 of [Lack and Sobociński, 2005]. □

The subtle point is that one obtains a "well-behaved" join of $\mathscr{M}$-subobjects (from the pushout over their intersection) although the resulting subobject is not necessarily again an $\mathscr{M}$-subobject. In a similar way, also directed unions are "well-behaved".

◌ PROPOSITION 7.19 (Directed unions and distributivity) Let $\mathbb{C}$ be a weakly $\mathscr{M}\omega$-adhesive category, and let $\{[a_i] \sqsubseteq [a_{i+1}]\}_{i\in\mathbb{N}}$ be a chain of $\mathscr{M}$-subobjects.

Then the join $\bigsqcup_{i\in\mathbb{N}}[a_i]$ of all $[a_i]$ exists at least in the poset $\mathrm{Sub}(Z)$ of arbitrary subobjects, and it can be computed via ω-composition of the chain $\{[a_i] \sqsubseteq [a_{i+1}]\}_{i\in\mathbb{N}}$. Moreover, the equation

$$c \sqcap \bigsqcup_{i\in\mathbb{N}} a_i = \bigsqcup_{i\in\mathbb{N}} (c \sqcap a_i)$$

holds for any arbitrary subobject $[c] \in \mathrm{Sub}(Z)$.

Proof. This proposition is a direct consequence of Lemma B.8. □

This list of facts about universal hereditary pushout categories forms the background for the following comparison of the variants of (weak) adhesivity.

Variations of adhesivity

8

While the previous section showed that most properties of adhesive categories that were presented in [Lack and Sobociński, 2005] can be adapted to universal hereditary pushout categories w.r.t. to a dominion $\mathscr{M}$ and hence also to weakly $\mathscr{M}$-adhesive categories, the present section completes the picture by including the weak adhesive HLR-categories of [Ehrig and Prange, 2006]. In fact, the major part of this section focuses on the relation between weak adhesive HLR-categories w.r.t. a suitable class $\mathscr{M}$ of monomorphisms on the one hand, and weakly $\mathscr{M}$-adhesive categories on the other hand; moreover, the latter two notions are shown to be equivalent for a wide range of categories.

The purpose of this section is to substantiate that adhesivity implies weak adhesivity and that each weakly $\mathscr{M}$-adhesive category is a weak adhesive HLR-category w.r.t. $\mathscr{M}$, and that the difference between the latter two classes of categories is only marginal. In other words, the main goal is to establish the following proposition.

PROPOSITION 8.1 Adhesivity hierarchy

1. Every adhesive category is weakly $\mathscr{M}ono$-adhesive.

2. Every weakly $\mathscr{M}$-adhesive category is a weak adhesive HLR-category w.r.t. $\mathscr{M}$.

3. Let $\mathbb{C}$ be a weak adhesive HLR-category w.r.t. $\mathscr{M}$. Then this category $\mathbb{C}$ is weakly $\mathscr{M}$-adhesive if there exist an $(\mathscr{E},\mathscr{M})$-factorization system for some class $\mathscr{E} \subseteq \mathscr{E}pi \subseteq \mathrm{ar}(\mathbb{C})$ of epimorphisms such that $\mathscr{E}$ is stable under pullback along $\mathscr{M}$-morphisms.

Overview of the section. This section mainly consists in giving the proof of this proposition. The proposition first makes precise in what sense weakly adhesive categories are a generalization of adhesive categories. Moreover, the proposition shows that the variation of adhesive categories which is proposed in this thesis coincides with the weak HLR-categories of [Ehrig and Prange, 2006] in many cases (though by definition it would be possible that there could exist categories which are weak HLR-categories but not weakly adhesive). The resulting hierachy of variations of adhesivity (including the universal hereditary pushout categories of Section 7 as an alternative to adhesive categories) is sketched in Figure 70.

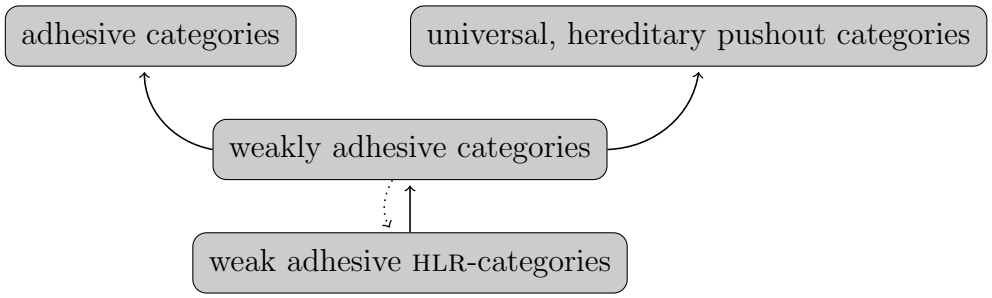


Figure 70: Adhesivity hierarchy (arrows indicate specialization)

From adhesivity to weak adhesivity. The starting point is the fact that all adhesive categories are weakly adhesive. As already mentioned above, (proper) Van Kampen squares as described in Definition 2.20 are trivially $\mathscr{M}ono$-universal where $\mathscr{M}ono$ is the class of all monomorphisms. However VK squares also satisfy the $\mathscr{M}ono$-pullback merging property.

LEMMA 8.2 (Van Kampen squares merge $\mathscr{M}ono$-pullback complements) Let $\mathbb{C}$ be a category with pullbacks, let $B \leftarrow f - A - m \rightarrow C$ be a $\mathbb{C}$-span, and further let $B - n \rightarrow D \leftarrow g - C$ be a pushout of $B \leftarrow f - A - m \rightarrow C$, such that the pushout square $\begin{smallmatrix} A & \rightarrow & C \\ \downarrow & & \downarrow \\ B & \rightarrow & D \end{smallmatrix}$ is a proper Van Kampen square.

Then the pushout $B - n \rightarrow D \leftarrow g - C$ merges $\mathscr{M}ono$-pullback complements.

Proof. Assume that the square $\begin{smallmatrix} B & \leftarrow & A \\ & \searrow & \\ & D & \leftarrow & C \end{smallmatrix}$ is a Van Kampen square; further consider the middle one of the three diagrams in Figure 71 over the latter pushout square. Assume that the top face $\begin{smallmatrix} B' & \leftarrow & A' \\ & \searrow & \\ & D' & \leftarrow & C' \end{smallmatrix}$ is a pushout square and

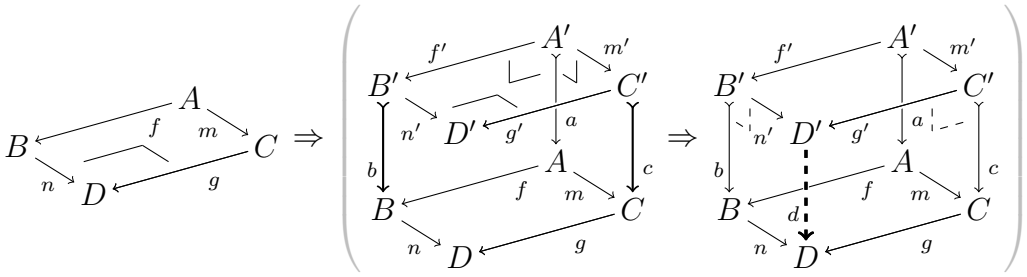


Figure 71: First part of the proof of Lemma 8.2

that the back faces $\begin{smallmatrix} B' & \leftarrow & A' \\ \downarrow & & \downarrow \\ B & \leftarrow & A \end{smallmatrix}$ and $\begin{smallmatrix} A' & \rightarrow & C' \\ \downarrow & & \downarrow \\ A & \rightarrow & C \end{smallmatrix}$ are pullback squares.

Then by the universal property of the top pushout, there is a unique mediating morphism $d\colon D' \to D$ satisfying $b \fatsemi n = n' \fatsemi d$ and $g \circ c = d \circ g'$

(see the right hand diagram in Figure 71). Further, as the square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$ was assumed to be Van Kampen, the front faces ${}^{B'}_{B}\downarrow\overrightarrow{\square}\downarrow{}^{D'}_{D}$ and ${}^{D'}_{D}\downarrow\overleftarrow{\square}\downarrow{}^{C'}_{C}$ are pullback squares; it remains to show that d is monic.

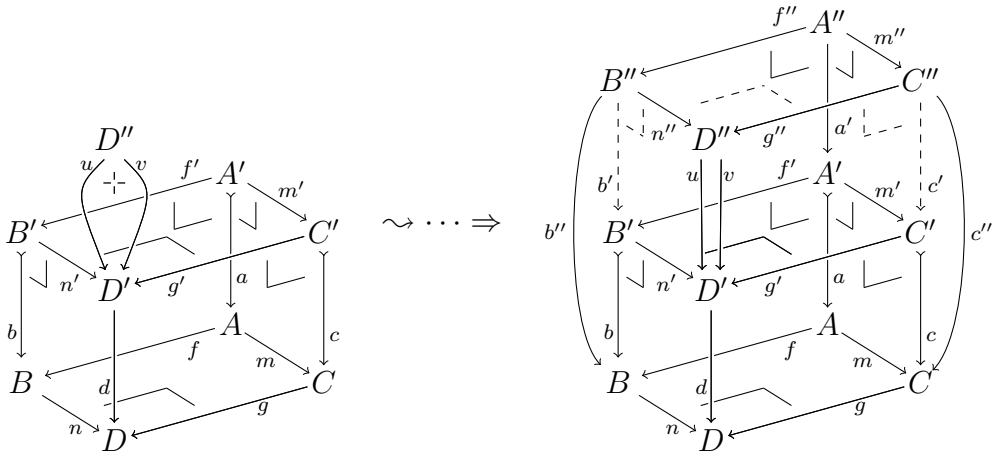


Figure 72: Proof sketch of Lemma 8.2

To show this, let $u, v\colon D'' \to D'$ be two morphisms satisfying the equation $u \fatsemi d = v \fatsemi d =: k$. Now construct two pullbacks along k, namely the pullbacks $B \leftarrow b''- B'' -n''\rightarrow D''$ and $C \leftarrow c''- C'' -g''\rightarrow D''$, of the co-spans $B -n\rightarrow D \leftarrow k- D''$ and $C -g\rightarrow D \leftarrow k- D''$, respectively (see the right hand diagram in Figure 72). Further, the morphisms $b'\colon B'' \to B'$ and $c'\colon C'' \to B''$ arise by pullback splitting of the large front pullback rectangles of the right hand double cube diagram in Figure 72. Finally the shown double cube is completed by the pullback $B'' \leftarrow f''- A'' -a'\rightarrow A'$ which gives rise to the pullback square ${}^{B''}_{B'}\downarrow\overleftarrow{\square}\downarrow{}^{A''}_{A'}$, and – using pullback splitting once more – there is a morphism $m''\colon A'' \to C''$ such that ${}^{A''}_{A'}\downarrow\overrightarrow{\square}\downarrow{}^{C''}_{C'}$ is a pullback square.

The pushout of the bottom Van Kampen square is universal by definition, and hence the top face is a pushout, which implies that both u and v are equal to the unique mediating morphism $w\colon D'' \to D'$ satisfying $w \circ g'' = g' \circ c'$ and $n'' \fatsemi w = b' \fatsemi n'$. □

Corollary 8.3 Every adhesive category is weakly $\mathscr{M}ono$-adhesive.

From weakly adhesive to weak adhesive HLR. The next task is to show that each weakly $\mathscr{M}$-adhesive category is also a weak adhesive HLR category w.r.t. $\mathscr{M}$. The central proposition is that that pushouts along pairs of

$\mathscr{M}$-morphisms are not only universal but also merge $\mathbb{C}$-pullback complements. In other words, pushouts along pairs of $\mathscr{M}$-morphisms are *proper* Van Kampen squares.

PROPOSITION 8.4 (Van Kampen squares from universality) In a weakly $\mathscr{M}$-adhesive category, pushouts along pairs of $\mathscr{M}$-morphisms join $\mathrm{ar}(\mathbb{C})$-pullback complements (and hence are properly Van Kampen).

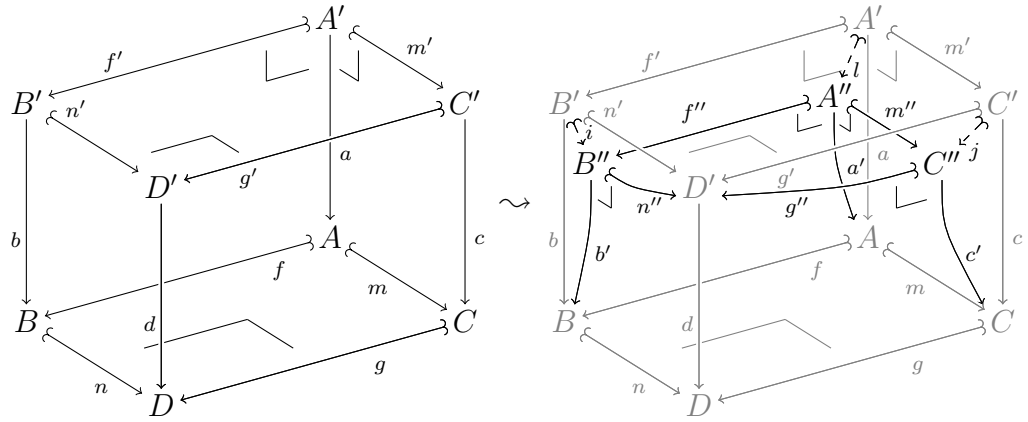


Figure 73: Proof sketch of Proposition 8.4

Proof. Given a cube diagram as shown on the left in Figure 73, take the pullback of the bottom pushout square ${}^{B\leftarrow A\searrow}_{\searrow D\leftarrow C}$ along $d\colon D' \to D$, yielding a new "inner" cube with top pushout ${}^{B''\leftarrow A''\searrow}_{\searrow D'\leftarrow C''}$ which is "inside" the original cube, i.e. using the universal properties of pullbacks there are unique morphisms $B' \hookleftarrow i \hookrightarrow B''$, $C' \hookleftarrow j \hookrightarrow C''$, and $A' \hookleftarrow l \hookrightarrow A''$ as illustrated on the right in Figure 73.

This give rise to a new cube on top with ${}^{B''\leftarrow A''\searrow}_{\searrow D'\leftarrow C''}$ as bottom face, and ${}^{B'\leftarrow A'\searrow}_{\searrow D'\leftarrow C'}$ as top face, where the "vertical" arrow from D to D is the identity. Using the fact that $\mathbb{C}$ is in a weakly $\mathscr{M}$-adhesive category, the pushout ${}^{B''\leftarrow A''\searrow}_{\searrow D'\leftarrow C''}$ is converse mono-universal, and as its back faces are pullbacks by pullback splitting, the upper two front "triangles-squares" from $B' \hookleftarrow i \hookrightarrow B''$ to $D' \xrightarrow{\mathrm{id}} D'$ and from $C' \hookleftarrow j \hookrightarrow C''$ to $D' \xrightarrow{\mathrm{id}} D'$ are pullbacks, which in turn means that all three of i, j, and l are invertible. □

This proposition (together with 7.9) implies that each weakly $\mathscr{M}$-adhesive category is a weak adhesive HLR-category w.r.t. $\mathscr{M}$.

From weak adhesive HLR *to weakly adhesive.* Recall that a class of monomorphisms $\mathscr{M}$ is called *admissible* if it contains all identities, and is closed under composition and pullbacks, i.e. $(B \xleftarrow{} C), (A \xleftarrow{} B) \in \mathscr{M}$ implies $m \circ n \in \mathscr{M}$ and for any pullback $A \xleftarrow{n} N \xrightarrow{g} M$ of a pullback square $\begin{smallmatrix} A & \downarrow & \leftarrow \\ D & \leftarrow & \downarrow \\ & & N \\ & & M \end{smallmatrix}$ arising from some cospan $A \xrightarrow{f} D \xleftarrow{m} M$ with $m \in \mathscr{M}$, we have $n \in \mathscr{M}$.

To show the last part of the adhesivity hierarchy, some auxiliary properties are derived first. Assume that the category $\mathbb{C}$ has an $(\mathscr{E}, \mathscr{M})$ factorization system (see Definition A.32) where $\mathscr{M}$ is as in Definition 2.45, and $\mathscr{E}$ is the class of epimorphisms. Hence every $\mathbb{C}$-morphism d factors into $e \in \mathscr{E}$, followed by $d' \in \mathscr{M}$. Moreover, assume that the class $\mathscr{E}$ is stable under pullback (along $\mathscr{M}$-morphisms).

LEMMA 8.5 (Pushout complements) If $\mathbb{C}$ is a weak adhesive HLR category w.r.t. $\mathscr{M}$ with a stable $(\mathscr{E}, \mathscr{M})$-factorization system, then pushout complements are ar($\mathbb{C}$)-final pullback complements, i.e. for every pushout square $\begin{smallmatrix} M & \downarrow & \leftarrow \\ B & \leftarrow & \downarrow \\ & & A \\ & & N \end{smallmatrix}$ based on a span $N \xleftarrow{f} M \xleftarrow{m} A$ with $m \in \mathscr{M}$ and a pushout $N \xleftarrow{n} B \xleftarrow{g} A$, there is a natural isomorphisms of homsets $\vartheta_x \colon \mathbb{C}{\downarrow}A(g^*(x), m) \xrightarrow{\cong} \mathbb{C}{\downarrow}B(x, n)$ where $g^* \colon \mathbb{C}{\downarrow}B \to \mathbb{C}{\downarrow}A$ is a any choice of a functor which acts by pulling back along g, i.e. $\Sigma_g \dashv g^*$.

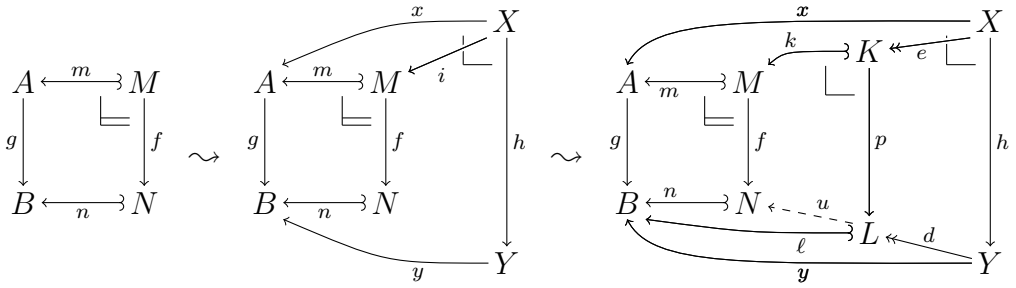


Figure 74: Illustration of the proof of Lemma 8.5

Proof sketch. The proof of Lemma 2.8 of [Lack and Sobociński, 2005] can be adapted to obtain a natural isomorphism

$$\psi_x \colon \operatorname{Sub}_{\mathscr{M}}(C)(g^{-1}(x), [m]) \xrightarrow{\cong} \operatorname{Sub}_{\mathscr{M}}(D)(x, [n]).$$

Now the result follows from the existence and stability of $(\mathscr{E}, \mathscr{M})$-factorizations as illustrated in Figure 74. □

A corollary of this lemma is the so-called special pullback-pushout property (see [Ehrig and Kreowski, 1979]), which will be used below.

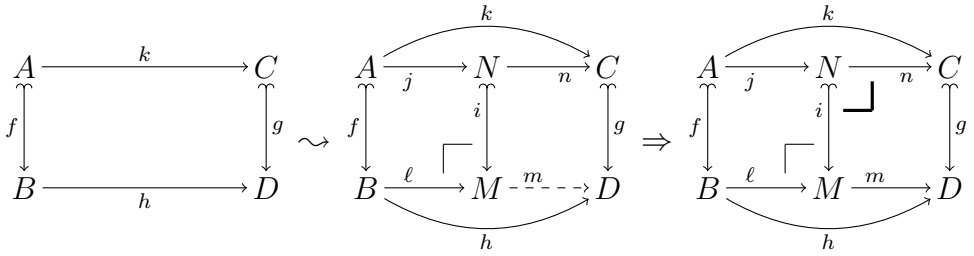


Figure 75: Special pullback-pushout property

COROLLARY 8.6 (Special pulllback-pushout property) If $\mathbb{C}$ is a weak adhesive HLR category w.r.t. $\mathscr{M}$ with stable $(\mathscr{E},\mathscr{M})$-factorizations then the special pullback-pushout property holds (see Figure 75).

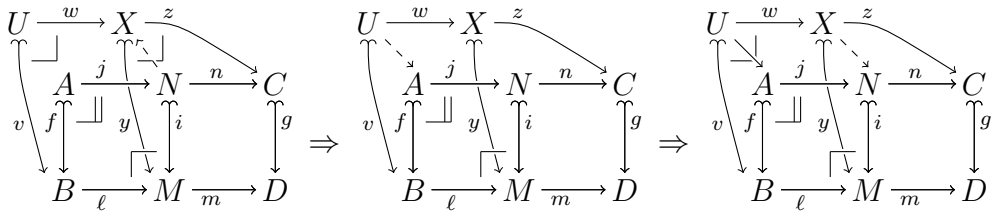


Figure 76: Proof idea of the special pullback-pushout property

Proof sketch. As illustrated in Figure 76, let $M \leftarrow y \rightarrowtail X \rightarrow z \rightarrow C$ be the pullback of $M \rightarrow m \rightarrow D \leftarrow g \rightarrowtail C$, and further let $B \leftarrow v \rightarrowtail U \rightarrow w \rightarrow X$ be the pullback $B \rightarrow l \rightarrow M \leftarrow y \rightarrowtail X$. Now the inclusion $i \sqsubseteq y$ holds in $\mathrm{Sub}_{\mathscr{M}} M$, since $M \leftarrow y \rightarrowtail X \rightarrow z \rightarrow C$ is a pullback of $M \rightarrow m \rightarrow D \leftarrow g \rightarrowtail C$, and $m \circ i = g \circ n$ by assumption. Hence it remains to show that also $y \sqsubseteq i$ holds. However, using that the front rectangle is a pullback by assumption, conclude that there must be a mediating morphism $U \to A$, which in turn yields a morphism $X \to N$ that witnesses $v \sqsubseteq f$ whence $y \sqsubseteq i$ by Lemma 8.5. □

Now all auxiliary results are established to finish the proof of Proposition 8.1.

LEMMA 8.7 Let $\mathbb{C}$ be a weak adhesive HLR category w.r.t. $\mathscr{M}$. Further let $\mathbb{C}$ have a stable $(\mathscr{E},\mathscr{M})$ factorization system, where $\mathscr{E}$ is a class of epimorphisms.

Then pushouts along $\mathscr{M}$-morphisms in $\mathbb{C}$ merge $\mathscr{M}$-pullback complements.

Proof sketch. Starting with a cube diagram as shown on the left in Figure 77, factor the morphism d into $e \in \mathscr{E}$ and $d' \in \mathscr{M}$ and let D'' be the domain of d'.

Next pull the bottom pushout square $\begin{smallmatrix} B \leftarrow A \searrow \\ \searrow D \leftarrow C \end{smallmatrix}$ back along $d' \colon D'' \to D$; this yields a new "inner" cube with the pushout square $\begin{smallmatrix} B'' \leftarrow A'' \searrow \\ \searrow D'' \leftarrow C'' \end{smallmatrix}$ on top which is "inside" the original cube, i.e. using the universal properties of pullbacks there are unique morphisms $B' \hookleftarrow i \hookrightarrow B''$, $C' \hookleftarrow j \hookrightarrow C''$, and $A' \hookleftarrow l \hookrightarrow A''$ as illustrated on the right in Figure 77. Now, using the special pullback-pushout property,

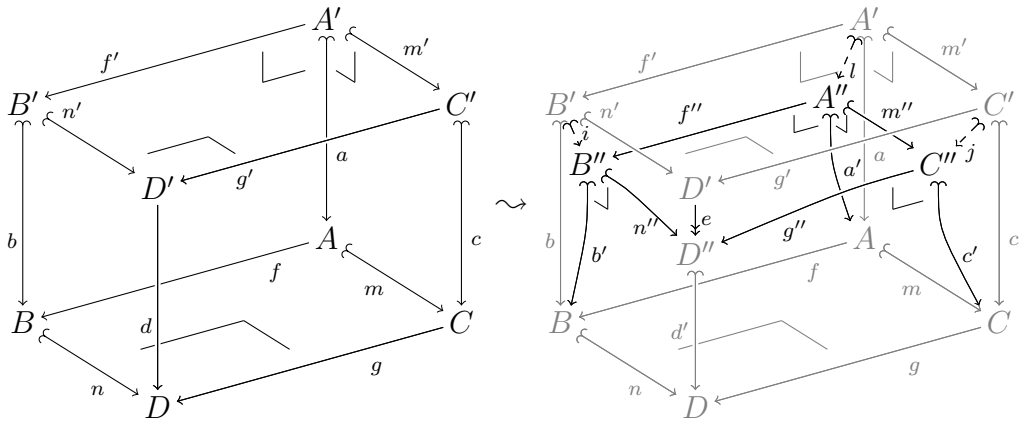


Figure 77: Proof sketch of Lemma 8.7

derive that i is the image of e under pullback along n''. Hence, i is not only a regular monic epic and thus must be and isomorphism, which in turn implies that also l is invertible. Finally, using pushout splitting, conclude that the right rectangle in front is a pushout square, i.e. $D' \hookrightarrow_e D'' \hookleftarrow_{g''} C''$ is the pushout of $D' \hookleftarrow_{g'} C' \hookrightarrow_j C''$. □

9 *Weak adhesivity from a topos theoretic perspective*

This final section on (weak) adhesivity discusses in what sense these categories are set-like. This approach is influenced by the papers [Johnstone et al., 2007, Lack and Sobociński, 2006] and the following general idea.

> One can think of adhesive categories as categories in which pushouts along monomorphisms are "well-behaved", where the paradigm for behaviour is given by the category of sets.
>
> [Lack and Sobociński, 2005, page 2]

Hence the starting point is the review of this paradigm category of sets and functions. However, the angle of approach is category theoretical and is based on an established abstraction of the category of sets and functions, namely the notion of a topos [Johnstone, 1977], which will be reviewed in Section 9.1.

The claim is that to understand (weak) adhesivity it suffices to understand the behaviour of colimits in any (quasi-)topos. Hence, the first goal is to make precise, in what sense pushouts (along monomorphisms) are "well-behaved" in a topos. The presentation focuses on the two properties of pushouts along (pairs) of $\mathscr{M}$-morphisms in the definition of weakly $\mathscr{M}$-adhesive categories, namely universality and stability w.r.t. the embedding into the category of $\mathscr{M}$-partial maps. In fact, the first part of this section elaborates on the fact that *every* colimit in any topos enjoys these properties – and not only pushouts of pairs of monomorphisms. Moreover, the first theorem of this section is that a topos is weakly ω-adhesive if and only if it has countable coproducts.

However, there are many examples of categories which are (weakly) adhesive but not a topos. As paradigmatic example, take the category of topological spaces $\mathbb{T}\mathbb{o}\mathbb{p}$, which provides the intuition for the informal description of "gluing" and "clipping" in the introduction. In $\mathbb{T}\mathbb{o}\mathbb{p}$, all colimits exist and are preserved by the graphing functor into suitable partial map categories; however not even all pushouts are universal. Hence universality of colimits is not shared by all "graph-like" structures (cf. [Prange, 2007, Example 1]).

Taking the examples of topoi and topological spaces together suggests that one could slightly strengthen weak $\mathscr{M}$-adhesivity by requiring the existence of all pushouts and stability w.r.t. the embedding into the category of $\mathscr{M}$-partial maps. The results of such a strengthening have been discussed in Section 7; the study of related notions of adhesivity was given separately in Section 8. The reason for this decision is that the latter two topics are fairly independent

of the present topos-theoretic perspective, which presupposes some experience with category theory.

Overview of the section. The main purpose of this section is to present the fundamental facts about topoi which allow to show that they are weakly $\mathscr{M}\!\mathit{ono}$-adhesive. The starting point is the definition of topos, which is given in Section 9.1 in the style of a category theoretical abstraction of the category of sets and functions. The two main topics are universality of colimits and partial map classification. One might want to think of universality as a generalization of the phenomenon that pre-images of unions are the same as the unions of the pre-images; further partial map representation is the category theoretical counterpart of the well-known practice to identify partial functions with total functions that yield a designated undefined element whenever the function value is not defined (in the original partial function). The fact that each topos is a weakly $\mathscr{M}\!\mathit{ono}$-adhesive category then follows as a direct consequence of universality of colimits and the existence of partial map classifiers.

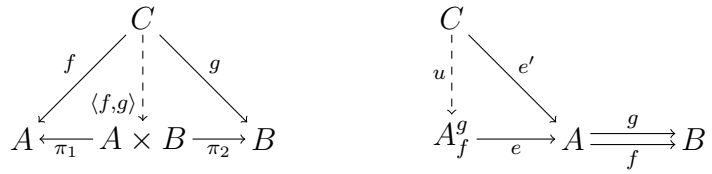
Finally, Section 9.2 addresses the paradigmatic example of a category that is weakly $\mathscr{M}\omega$-adhesive even though it is not a (quasi-)topos, namely the category of topological spaces $\mathbb{T}\mathbb{o}\mathbb{p}$. The crucial fact is that not all colimits are universal in $\mathbb{T}\mathbb{o}\mathbb{p}$ and this is also the reason why $\mathbb{T}\mathbb{o}\mathbb{p}$ is not a universal hereditary pushout category. Hence, in the end, though the requirement of universality of *all* colimits seems natural in the context of topos theory, it would rule out some example categories (see also [Prange, 2007, Example 1]).

9.1 A topos theory primer

The notion of topos is taken as the paradigmatic generalization of the category sets and functions. One of the fundamental ideas of topos theory is that any topos allows to mimic the following familiar set theoretic constructions: first, the product $A \times B$ of two sets A and B is again a set (which contains all ordered pairs $\langle a, b\rangle$ such that $a \in A$ and $b \in B$ hold); second, there exists some singleton set $\{\star\}$ (which contains exactly one element); third, given any pair of parallel functions $f, g\colon A \to B$ there exists a subset $A_f^g \subseteq A$ on which f and g are equal, namely $A_f^g = \{a \in A \mid f(a) = g(a)\}$; fourth, given two sets A and B, all functions between from A to B form yet another set, namely $A^B = \{f \mid f\colon A \to B \text{ is a function}\}$; finally, there is a bijective correspondence between subsets of a given set A and all functions in $\{0,1\}^A$ such that each subset $A' \subseteq A$ can be identified with its characteristic function

$\chi_A \colon A \to \{0,1\}$.

☼ DEFINITION 9.1 (Topos) A category $\mathbb{C}$ is a *topos* if the following hold.



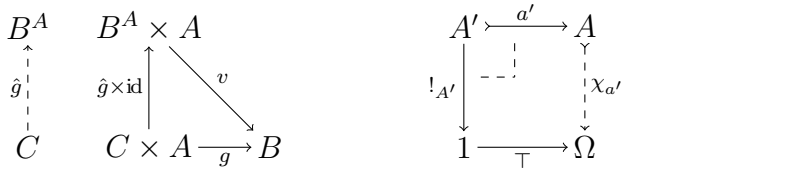
(a) Product of A and B (b) Equalizer of f and g

Figure 78: Products and equalizers

PRODUCTS EXIST For any pair of objects $A, B \in \mathbb{C}$, there exists an object $A \times B \in \mathbb{C}$ with two projection morphisms $\pi_1 \colon A \times B \to A$ and $\pi_2 \colon A \times B \to B$ such that for any pair of morphisms $f \colon C \to A$ and $g \colon C \to B$ with common domain C, there exists a unique morphisms $\langle f, g \rangle \colon C \to A \times B$ which satisfies the two equations $\pi_1 \circ \langle f, g \rangle = f$ and $\pi_2 \circ \langle f, g \rangle = g$ (see Figure 78(a)).

A TERMINAL EXISTS There exists an object $1 \in \mathbb{C}$ such that for every object $A \in \mathbb{C}$ there is exactly one arrow $!_A \colon A \to 1$.

EQUALIZERS EXIST For any pair of parallel morphisms $f, g \colon A \to B$, there exists a morphism $e \colon A^g_f \to A$ which equalizes f and g in a universal way, i.e. the equation $f \circ e = g \circ e$ holds and for any other morphism $e' \colon C \to A$ which satisfies $f \circ e' = g \circ e'$, there exists a unique morphism $u \colon C \to A^g_f$ such that $e' = e \circ u$ (see Figure 78(b)).



(a) Exponential B^A (b) Subobject classifier $\top \colon 1 \to \Omega$

Figure 79: Exponentials and subobject classifiers

EXPONENTIALS EXIST For each object $A \in \mathbb{C}$, the functor $\_ \times A \colon \mathbb{C} \to \mathbb{C}$, which maps $C \in \mathbb{C}$ to $C \times A$, has a right adjoint. This means that for any pair of objects $A, B \in \mathbb{C}$ there exists an exponential object B^A with an evaluation map $v \colon B^A \times A \to B$ such that for any object $C \in \mathbb{C}$ and any morphism $g \colon C \times A \to B$ there is a unique morphism $\hat{g} \colon C \to B^A$ such that $g = v \circ (\hat{g} \times \mathrm{id}_A)$ where $\hat{g} \times \mathrm{id}_A$ is the map $\langle \hat{g} \circ \pi_1, \mathrm{id}_A \circ \pi_2 \rangle \colon C \times A \to B^A \times A$ (see Figure 79(a)).

A SUBOBJECT CLASSIFIERS EXISTS There exists an object Ω with and arrow $\top \colon 1 \to \Omega$ such that for every monomorphism $a' \colon A' \rightarrowtail A$ there is a unique arrow $\chi_{a'} \colon A \to \Omega$ such that $1 \leftarrow !_{A'} - A' \rightarrowtail a' \rightarrow A$ is a pullback of $1 - \top \rightarrow \Omega \leftarrow \chi_{a'} - A$, thus yielding a pullback square $\begin{smallmatrix} A' & \downarrow & \top \\ 1 & \rightarrow & \Omega \end{smallmatrix}$ (see Figure 79(b)).

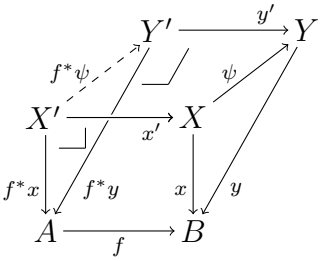
☼

As a first trivial consequence, any topos has all finite limits, in particular pullbacks. A non-trivial result is that topoi also have all finite colimits, and indeed the first definition of topos explicitly required the existence of all finite colimits [Johnstone, 1977]. However, the two central properties of topoi in the context of this thesis are the representability of partial maps as total maps and the universality of all colimits, as discussed next.

Universality of colimits. A typical instance of the universality of colimits in the category of sets is the equation $f^{-1}(X \cup Y) = f^{-1}(X) \cup f^{-1}(Y)$ where $f \colon A \to B$ is a function, $f^{-1} \colon \wp(B) \to \wp(A)$ is the pre-image operation, and X and Y are subsets of B. The general case is obtained by replacing the subsets $X, Y \subseteq B$ by arbitrary morphisms $x \colon X \to B$ and $y \colon Y \to B$ in a category $\mathbb{C}$ with pullbacks; further the pre-image operation $f^{-1} \colon \wp(B) \to \wp(A)$ becomes a pullback functor $f^* \colon \mathbb{C}{\downarrow}B \to \mathbb{C}{\downarrow}A$ between slice categories.

☼ DEFINITION 9.2 (Pullback functor) Let $\mathbb{C}$ be any category, let $A, B \in \mathbb{C}$ be two objects, and let $f \colon A \to B$ be a morphism in $\mathbb{C}$.

Then a *pullback functor (based on f)* is a functor $f^* \colon \mathbb{C}{\downarrow}B \to \mathbb{C}{\downarrow}A$ which acts by pulling back along the morphism $f \colon A \to B$ as illustrated to the right, i.e. for each object $(X - x \rightarrow B) \in \mathbb{C}{\downarrow}B$ there exists a pullback $A \leftarrow x_f - X' - x' \rightarrow X$ of $A - f \rightarrow B \leftarrow x - X$ in $\mathbb{C}$ which satisfies $x_f = f^*(x)$ and moreover, given a choice of such pullbacks, $y' \circ f^*(\psi) = \psi \circ x'$ holds for all morphisms $\psi \colon x \to y$ in $\mathbb{C}{\downarrow}B$. The objects X' and Y' are often written as $f^*(X)$ and $f^*(Y)$, respectively. ☼



The fundamental theorem for topoi [Freyd, 1972, Theorem 2.31] states that pullback functors have not only left but also right adjoints, and hence pullback functors transfer colimits of one slice category to another one. Using this fact, universality of colimits in topoi directly follows from the existence of products as spelled out in Lemma 9.4 below.

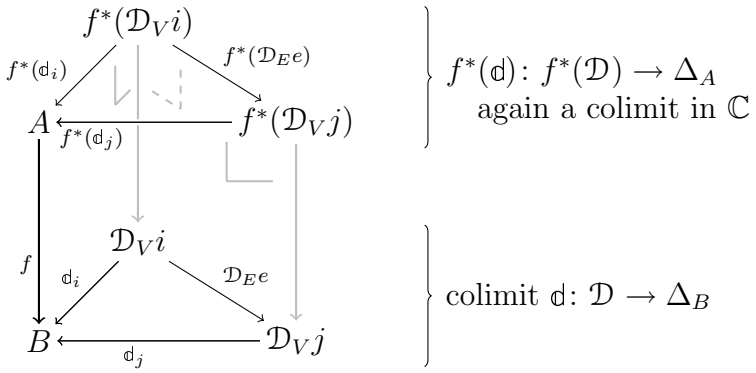


Figure 80: Universality/pullback-stability of colimits

Universality of colimits is illustrated in Figure 80: if the bottom face is a universal colimit in $\mathbb{C}$, then pulling back the diagram along f as indicated yields again a colimit in $\mathbb{C}$. Universal colimits are also referred to as pullback-stable colimits. More formally, they can be defined as follows.

DEFINITION 9.3 (Universality of colimits) Let $\mathbb{C}$ be any category, let $\mathcal{D}\colon \mathbb{I} \to \mathbb{C}$ be a diagram and let $\mathbb{d}\colon \mathcal{D} \to \Delta_B$ be a colimit for $\mathcal{D}$; further let $\mathcal{D}'\colon \mathbb{I} \to \mathbb{C}{\downarrow}B$ be the induced diagram in the slice category $\mathbb{C}{\downarrow}B$ which satisfies $\mathcal{D}'_V(i) = \mathbb{d}_i$ for all $i \in \mathbb{I}$ and $|\mathcal{D}'_E(e)|_B = \mathcal{D}_E(e)$ for all arrows $e\colon i \to j$ in $\mathbb{I}$.

Then the colimit $\mathbb{d}$ is *universal* if for any pullback functor $f^*\colon \mathbb{C}{\downarrow}B \to \mathbb{C}{\downarrow}A$ the cocone $\{f^*(\mathbb{d}_i)\}_{i\in\mathbb{I}}\colon f^*(\mathcal{D}) \to \Delta_A$ is again a colimit where $f^*(\mathcal{D})$ is the diagram $|\_|_A \circ f^* \circ \mathcal{D}'$ (see also Figure 80).

That universality of colimits is a non-trivial property is witnessed by the counterexample of Figure 81, which is taken from [Lack and Sobociński, 2005, Example 3.4]. It shows that the pushout of the two different morphisms of $\{\!|1|\!\}$ into $\{\!|2|\!\}$ is not stable under pullback along the morphism from $\{\!|2|\!\}$ to the pushout object $\{\!|3|\!\}$ which maps 0 and 1 to 0 and 2, respectively. However, in any topos all colimits are universal – a fact which is a consequence of the fundamental theorem for topoi and the following basic lemma about products.

LEMMA 9.4 Let $\mathbb{C}$ be a category with pullbacks, let $A \in \mathbb{C}$ be an object, let $\mathcal{D}\colon \mathbb{I} \to \mathbb{C}\downarrow A$ be a diagram such that the terminal $\mathrm{id}_A \in \mathbb{C}\downarrow A$ arises as

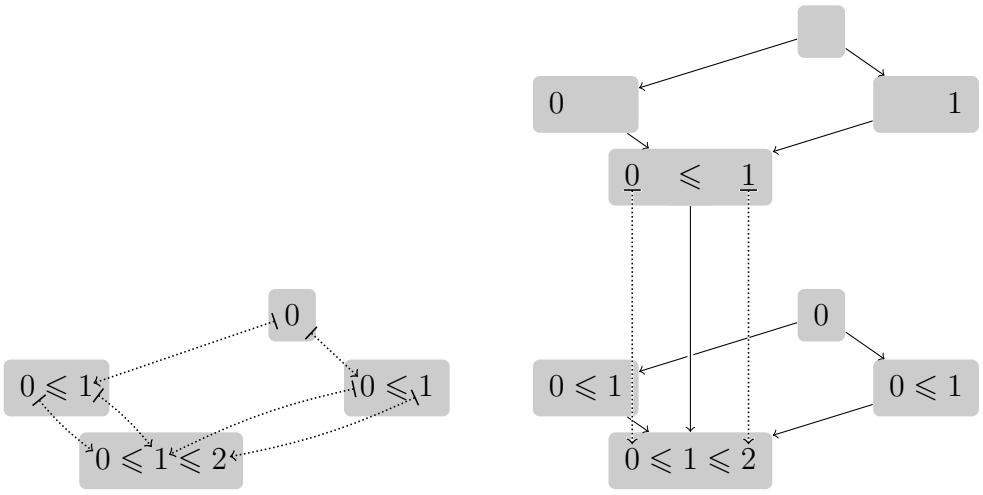


Figure 81: A non-universal pushout of posets

the colimit object of $\mathcal{D}$, i.e. such that the family $\mathbb{d} = \{!_{\mathcal{D}_V(i)} \colon \mathcal{D}_V(i) \to \mathrm{id}_A\}_{i \in \mathbb{I}}$ yields a colimit $\mathbb{d} \colon \mathcal{D} \to \Delta_{\mathrm{id}_A}$ in the slice category $\mathbb{C} \downarrow A$.

Then, provided that $\mathbb{C}$ has products, the forgetful functor $|\_|_A \colon \mathbb{C} \downarrow A \to \mathbb{C}$ preserves the colimit $\mathbb{d}$, i.e. $|\_|_A * \mathbb{d} = \{|\mathbb{d}_i|_A\}_{i \in \mathbb{I}}$ is a colimit of $|\_|_A \circ \mathcal{D}$ in the base category $\mathbb{C}$.

Proof. Writing $|\mathcal{D}|$ and $|\mathbb{d}|$ for $|\_|_A \circ \mathcal{D}$ and $|\_|_A * \mathbb{d}$, respectively, let $\mathbb{f} \colon |\mathcal{D}| \to \Delta_C$ be any other cocone. Now combine the cocones $|\mathbb{d}|$ and $\mathbb{f}$ by means of the universal property of the product $A \leftarrow \pi_1 - A \times C - \pi_2 \rightarrow C$ to obtain the cocone $\mathbb{g} := \{\langle |\mathbb{d}_i|, \mathbb{f}_i \rangle \colon |\mathcal{D}(i)| \to A \times C\}_{i \in \mathbb{I}}$ from $|\mathcal{D}|$ to $A \times C$, as illustrated in Figure 82. Hence, the cocone $\mathbb{g}$ induces a $\mathbb{C} \downarrow A$-cocone $\mathbb{g}' = \{\mathbb{g}_i\}_{i \in I} \colon \mathcal{D} \to \Delta_{\pi_1}$. The

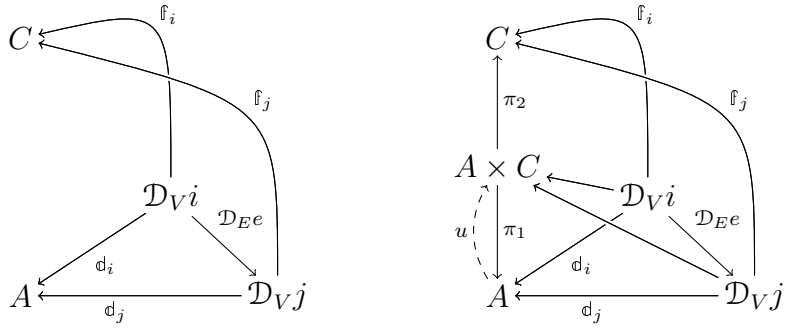


Figure 82: Proof sketch of Lemma 9.4

universal property of the colimit $\mathbb{d}$ yields a mediating morphism $u\colon \mathrm{id}_A \to \pi_1$ in $\mathbb{C}\downarrow A$, and as an immediate consequence $\pi_2 \circ |u|_A\colon A \to C$ is a mediating morphism from $|\mathbb{d}|$ to $\mathbb{f}$. Finally, that the latter morphism is the unique mediating morphism can be shown using the equation $\mathrm{id}_A = \pi_1 \circ |u|_A$ in $\mathbb{C}$. □

Corollary 9.5 (Universality of colimits) If a category $\mathbb{C}$ is a topos then all colimits that exist in $\mathbb{C}$ are universal.

Proof. Let $\mathbb{C}$ be a topos, let $\mathcal{D}\colon \mathbb{I} \to \mathbb{C}$ be a diagram and let $\mathbb{d}\colon \mathcal{D} \to \Delta_B$ be a colimit for $\mathcal{D}$; further let $\mathcal{D}'\colon \mathbb{I} \to \mathbb{C}\downarrow B$ be the induced diagram the slice category $\mathbb{C}\downarrow B$ which satisfies $\mathcal{D}'_V(i) = \mathbb{d}_i$ for all $i \in \mathbb{I}$ and $|\mathcal{D}'_E(e)|_B = \mathcal{D}_E(e)$ for all arrows $e\colon i \to j$ in $\mathbb{I}$. Finally let $f\colon A \to B$ be a morphism in $\mathbb{C}$, and let $f^*\colon \mathbb{C}\downarrow B \to \mathbb{C}\downarrow A$ be a pullback functor $\Sigma_h \dashv h^*$.

Now, the family $\{!_{\mathbb{d}_i}\}_{i\in\mathbb{I}}$ is a colimit for $\mathcal{D}'$ in the slice category $\mathbb{C}\downarrow B$. Hence, applying the fundamental theorem for topoi, which gives a right adjoint $\Pi_f \vdash f^*$, the family $\{f^*(\mathbb{d}_i)\}_{i\in\mathbb{I}}$ is a colimit of $f^*\circ\mathcal{D}'$ since f^* preserves colimits. Now apply Lemma 9.4 to establish $\mathbb{d}$ as a universal colimit. □

Partial map representation. In the category of sets, representability of partial maps is nothing else but the bijective correspondence between partial functions from A to B and all (total) functions from A to $B \uplus \{\bot\}$. More precisely there is a natural isomorphism $\mathbb{Pfn}(A,B) \cong \mathbb{Set}(A, B \uplus \{\bot\})$ for any sets A and B. As an equivalent formulation, the graphing functor $\Gamma\colon \mathbb{Set} \to \mathbb{Pfn}$, which maps each total function to the corresponding total function, has a right adjoint; the latter *lifting functor* maps each set B to $B \uplus \{\bot\}$ and each partial function $f\colon A \rightharpoonup B$ to the corresponding total $\bot$-preserving function $f_\bot\colon A \uplus \{\bot\} \to B \uplus \{\bot\}$ for which $f_\bot(a) = \bot$ holds for an element $a \in A$ if and only if $f(a)$ is undefined.

Generalizing from $\mathbb{Set}$ to some topos $\mathbb{C}$, the embedding of the category of sets into the category of partial functions becomes the graphing functor $\Gamma\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$, which maps every (total) morphism $f\colon A \to B$ to its graph $\Gamma_f = [\mathrm{id}_A, f\rangle\colon A \rightharpoonup B$, which is not to be confused with any object of the concrete category $\mathbb{Graph}$. Now, the *representation* or *classification* of partial maps in the topos $\mathbb{C}$ can be described as follows.

Fact 9.6 (Partial map representation) Let $\mathbb{C}$ be a topos. Then the graphing functor $\Gamma\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$ has a right adjoint.

To spell this out in detail, let $\mathcal{L}\colon \mathrm{Par}(\mathbb{C}) \to \mathbb{C}$ be the right adjoint $\Gamma \dashv \mathcal{L}$, and $\eta\colon \mathrm{id}_\mathbb{C} \to \mathcal{L}\circ\Gamma$ the unit of the adjunction, which embeds each object $B \in \mathbb{C}$ into its *lifting* $\mathcal{L}B$ as a monomorphism $\eta_B\colon B \rightarrowtail \mathcal{L}B$.

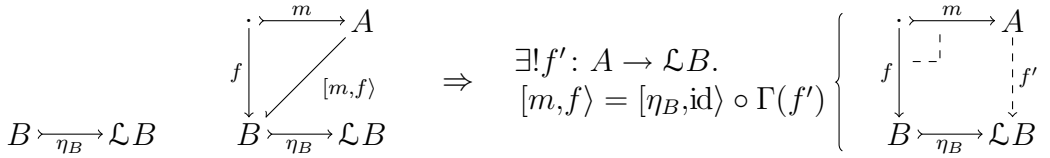


Figure 83: Representation of partial maps

Now, for each partial map $[m,f\rangle \colon A \rightharpoonup B$ there exists a unique morphism $f' \colon A \to \mathfrak{L}B$ such that the equation $[m,f\rangle = [\eta_B,\mathrm{id}\rangle \circ \Gamma(f')$ holds in $\mathrm{Par}(\mathbb{C})$ or (equivalently) such that $B \leftarrow f - \cdot \rightarrowtail m \rightarrow A$ is a pullback of $B \rightarrowtail \eta_B \rightarrow \mathfrak{L}B \leftarrow f - A$ (see Figure 83). The morphism $\eta_B \colon B \rightarrowtail \mathfrak{L}B$ is also known as the partial map classifier of B, and the subobject classifier $\top \colon 1 \rightarrowtail \Omega$ is the partial map classifier for the terminal object $1 \in \mathbb{C}$. The first relevant consequence of the existence of partial map classifiers is that all colimits that exist in some topos $\mathbb{C}$ are preserved by the embedding of the graphing functor $\Gamma \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$. Hence the following proposition holds.

Proposition 9.7 (Topoi are weakly adhesive) Every topos is weakly $\mathscr{M}ono$-adhesive where $\mathscr{M}ono$ is the class of all monomorphisms.

However, for trivial reasons, not all topoi are weakly ω-adhesive, as the category of finite sets, which is the prime example of a topos, does not have colimits of ω-chains of monomorphisms. A necessary and sufficient condition for weak ω-adhesivity of a topos is the existence of countable coproducts.

Theorem 9.8 (Topoi and ω-adhesivity) A topos is weakly ω-adhesive if and only if it has countable coproducts.

Proof sketch. Given a topos $\mathbb{E}$ with countable coproducts, colimits of ω-chains can be obtained by means of the usual construction using coproducts and co-equalizers. As all colimits are universal and preserved by the graphing functor, the topos $\mathbb{E}$ is weakly ω-adhesive.

Conversely, given a weakly ω-adhesive topos $\mathbb{D}$, the countable coproduct $\coprod_{i \in \mathbb{N}} A_i$ of any set $\{A_i \mid i \in \mathbb{N}\} \subseteq \mathrm{ob}(\mathbb{D})$ can be obtained by constructing the colimit of the chain $\{\coprod_{j<i} A_i \rightarrowtail \coprod_{j \leq i} A_i\}_{i \in \mathbb{N}}$. □

9.2 Further examples of weak adhesivity

The only item in the definition of topos that involves monomorphisms is the existence of subobject classifiers. Now, variations of topoi w.r.t. an admissible

class of monomorphisms $\mathscr{M}$ arise by weakening the conditions on subobject classifiers. Indeed, a quasi-topos is a category with all finite limits, exponentials and a *regular*-partial map classifier. In general the notion of $\mathscr{M}$-partial map classifier reads as follows.

Definition 9.9 ($\mathscr{M}$-partial map classifier) Let $\mathscr{M}$ be an admissible class of monomorphisms in a category $\mathbb{C}$ with a terminal object $1 \in \mathbb{C}$. Then a *$\mathscr{M}$-partial map classifier* is an object Ω with an arrow $\top\colon 1 \to \Omega$ such that for every $\mathscr{M}$-morphism $a'\colon A' \hookrightarrow A$ there is a unique arrow $\chi_{a'}\colon A \to \Omega$ such that $1 \leftarrow !_{A'} - A' \hookleftarrow a' \rightarrow A$ is a pullback of $1 \xrightarrow{\top} \Omega \xleftarrow{\chi_{a'}} A$, thus yielding a pullback square $\begin{smallmatrix} A' & \to & 1 \\ \downarrow & & \downarrow \top \\ A & \xrightarrow{\chi_{a'}} & \Omega \end{smallmatrix}$.

Since, as a fact [Wyler, 1991], also any quasi-topos has right adjoints to pullback functors, one can reuse the above arguments to show that each quasi-topos is a weakly quasi-adhesive category, i.e. a weakly $\mathscr{R}\mathscr{e}\mathscr{g}$-adhesive category where $\mathscr{R}\mathscr{e}\mathscr{g}$ is the class of all *regular* monomorphisms.

The case is slightly different for the category of topological spaces. The reason is that $\mathbb{T}\mathbb{o}\mathbb{p}$ does not have right adjoints to pullback functors even though it is complete and cocomplete. In particular not all pushouts are universal as witness by the pushout in Figure 81. However, $\mathbb{T}\mathbb{o}\mathbb{p}$ does have partial map classifiers for open and closed monomorphisms (see [Niefield, 1982, Adámek et al., 1990]). Hence all colimits that exist are preserved by the respective graphing functors. Nevertheless, pushouts of pairs of open (or closed) morphisms are universal. The reason for this are twofold: first, by the definition of continuous function, pre-images of open and closed sets are again open and closed, respectively; second, open and closed sets are closed under binary unions which are obtained by taking the pushout over their intersection very much like in the category of sets, i.e. $\mathbb{T}\mathbb{o}\mathbb{p}$ has $\mathscr{O}\mathscr{p}\mathscr{e}\mathscr{n}$- and $\mathscr{C}\mathscr{l}\mathscr{o}\mathscr{s}\mathscr{e}\mathscr{d}$-effective unions. Hence, the fact that the category of topological spaces is in fact an example of a weakly $\mathscr{M}$-adhesive category follows from the following lemma.

Lemma 9.10 (Universal effective unions) Let $\mathbb{C}$ be a category with pushouts and an admissible class of monomorphisms $\mathscr{M}$; further assume that $\mathbb{C}$ has $\mathscr{M}$-partial map classifiers and $\mathscr{M}$-effective unions.

Then pushouts of pairs of $\mathscr{M}$-morphisms are universal if and only if the pre-images of joins are the join of their pre-images, i.e. if the equation $f^{-1}(a \sqcup b) = f^{-1}(a) \sqcup f^{-1}(b)$ holds for all morphisms $f\colon X \to Y$ and all $\mathscr{M}$-subobjects $a, b \in \mathrm{Sub}_{\mathscr{M}}(Y)$ where $a \sqcup b$ is the join of a and b in $\mathrm{Sub}_{\mathscr{M}}(Y)$.

Proof sketch. First suppose that the equation $f^{-1}(a \sqcup b) = f^{-1}(a) \sqcup f^{-1}(b)$ holds for all morphisms $f\colon X \to Y$ and all $\mathscr{M}$-subobjects $a, b \in \mathrm{Sub}_{\mathscr{M}}(Y)$.

Given a span of $\mathscr{M}$-morphisms $A \leftarrow m \rightarrowtail U \leftarrowtail n \rightarrow B$, let $A \leftarrowtail a \rightarrow Z \leftarrow b \rightarrowtail B$ be its pushout, yielding the pushout square $\begin{smallmatrix} U & \rightarrowtail & B \\ A & \rightarrow & Z \end{smallmatrix}$, which is also a pullback square by Lemma 7.9 below. Now, $[a] \sqcup [b] = [\mathrm{id}_Z]$ in $\mathrm{Sub}_{\mathscr{M}}(Z)$; further, pulling back the above pushout square along any morphism $g\colon X \to Z$ yields again a pushout square, since the equation $g^{-1}([a] \sqcup [b]) = g^{-1}([a]) \sqcup g^{-1}([b])$ implies that $g^{-1}([a]) \sqcup g^{-1}([b]) = [\mathrm{id}_X]$.

The converse direction involves only straightforward calculations based on the Pullback Lemma (cf. A.14). □

The facts that have been mentioned so far allow to use established results of the literature to show that a given category is weakly adhesive. Moreover, there are several ways to obtain further examples of weakly adhesive categories by means of the following constructions.

Proposition 9.11 (Closure of ω-adhesivity) Let $\mathbb{C}$ and $\mathbb{D}$ be weakly $\mathscr{M}\omega$-adhesive categories. Then the following categories are again ω-adhesive:

- the product category $\mathbb{C} \times \mathbb{D}$;
- the slice category $\mathbb{C}{\downarrow}T$ for any $T \in \mathbb{C}$;
- the co-slice category $I{\downarrow}\mathbb{C}$ for any $I \in \mathbb{C}$;
- the functor category $[\mathbb{X}, \mathbb{C}]$ for any category $\mathbb{X}$;
- the Artin-Wraith gluing $\mathbb{C}{\downarrow}\mathcal{F}$, i.e. the comma category $\mathrm{id}_{\mathbb{C}}{\downarrow}\mathcal{F}$ for any functor $\mathcal{F}\colon \mathbb{D} \to \mathbb{C}$ that preserves pullbacks (along $\mathscr{M}$-morphisms).

Proof sketch. In each case pullbacks along $\mathscr{M}$-morphisms and the relevant colimits are constructed componentwise. □

Of particular importance is the construction of slice categories, as the latter were used passim in the first chapter of the thesis.

Summarizing, the examples of weakly $\mathscr{M}\omega$-adhesive categories abound. Topological spaces with their open and closed maps as natural classes of monomorphisms are one of the central categories in mathematics. The category of simple graphs, which occurs ubiquitously in computer science, is a quasi-topos and thus a universal, hereditary pushout category for the class of regular monomorphisms. These two examples are interesting in so far as they do not fit the original framework of (quasi-)adhesive categories [Lack and Sobociński, 2005, Johnstone et al., 2007].

Conclusion and summary of the second part

Apart from establishing all basic properties of weakly $\mathscr{M}\omega$-adhesive categories that have been used in (the proofs of) the results about processes and unfoldings of grammars in the first part, the second part studies adhesivity and related concepts. The motivation was to identify characteristic properties of the category of graphs which are sufficient to develop the theory of the concurrent semantics of graph rewriting of [Baldan, 2000] in an abstract framework. The focus was on two properties that are enjoyed by all colimits in the category graphs, namely universality and stability w.r.t. the graphing functor.

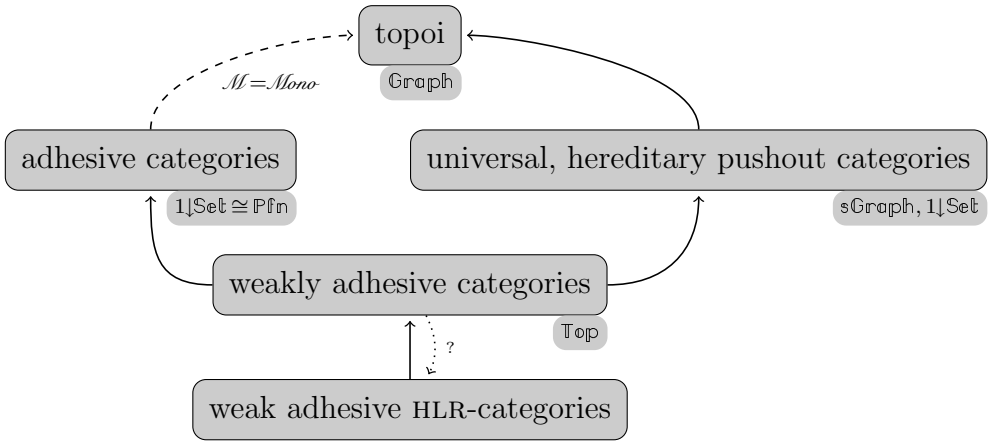


Figure 84: Extended adhesivity hierachy (arrows indicate specialization)

The resulting hierachy of "graph-like" categories is sketched in Figure 84;[79] the figure also gives typical examples for each class of categories. The strongest notion is that of a topos, which nevertheless is already general enough to account for the category of (multi-)graphs. However, adhesive categories, which are a standard framework for the theory of double pushout rewriting, encompass non-topos examples such as the category $1\downarrow\mathbb{S}et$, which happens to be isomorphic to the category of sets and partial functions $\mathbb{P}fn$.

As demonstrated in Section 7, universal, hereditary pushout categories share most of the good properties with adhesive categories and they have the advantage that *all* pushouts are sufficiently "well-behaved" . Moreover, their relation to topoi is robust under "parametrization" w.r.t. to a dominion $\mathscr{M}$. In contrast, as indicated by the dashed arrow in Figure 84, quasi-topoi are

[79] The dominion $\mathscr{M}$ is left implicit in Figure 84.

not quasi-adhesive in general [Johnstone et al., 2007]. The relevant example category is the category of simple graphs $\mathbb{sGraph}$, which is a quasi-topos (and thus a universal, $\mathscr{Reg}$-hereditary pushout category) but not a quasi-adhesive category.

Finally, weakly $\mathscr{M}$-adhesive categories are general enough to even include the category of topological spaces, which does not belong to any of the above mentioned classes. Based on the results of Section 8, the conjecture concerning weak adhesive HLR-categories w.r.t. to a class $\mathscr{M}$ is that there are no (application relevant) categories which belong to the latter class and are *not* weakly $\mathscr{M}$-adhesive. In other words – at present – it seems that weakly adhesive categories would be a viable alternative for the weak adhesive HLR-categories.

As a concluding remark, the (name of) partial Van Kampen colimits and squares of Section 7 could be approached from yet another angle. Replacing the category of *partial* maps (which generalizes partial functions) by the more general (bi-)category of *spans* (which one might want to think of as generalized relations) in the definition of partial Van Kampen colimit yields "proper" Van Kampen colimits (see the author's [Heindel and Sobociński, 2009]). As a consequence, the "proper" Van Kampen squares actually could be called *span* Van Kampen squares to distinguish them from the partial ones. The conclusion is that – glossing over some (bi-)categorical details – the central property of pushouts along monomorphisms in adhesive categories is very much the same as the defining property of partial Van Kampen colimits with spans in place of partial maps.

References

[ACETO AND INGÓLFSDÓTTIR, 2006] Aceto, L. and Ingólfsdóttir, A., editors (2006). *Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 25-31, 2006, Proceedings*, volume 3921 of *Lecture Notes in Computer Science*. Springer.

[ADÁMEK ET AL., 1990] Adámek, J., Herrlich, H., and Strecker, G. E. (1990). *Abstract and concrete categories: the joy of cats*. Wiley.

[BALDAN ET AL., 2004] Baldan, Corradini, and Konig (2004). Verifying finite-state graph grammars: An unfolding-based approach. In *CONCUR: 15th International Conference on Concurrency Theory*. Springer.

[BALDAN, 2000] Baldan, P. (2000). *Modelling Concurrent Computations: from Contextual Petri Nets to Graph Grammars*. PhD thesis, Dipartimento di Informatica, Università di Pisa.

[BALDAN ET AL., 2008A] Baldan, P., Bonchi, F., Heindel, T., and König, B. (2008a). Irreducible objects and lattice homomorphisms in adhesive categories. In Pfalzgraf, J., editor, *Proceedings of ACCAT '08 (Workshop on Applied and Computational Category Theory)*.

[BALDAN ET AL., 2008B] Baldan, P., Chatain, T., Haar, S., and König, B. (2008b). Unfolding-based diagnosis of systems with an evolving topology. In *Proceedings of CONCUR '08*, pages 203–217. Springer.

[BALDAN ET AL., 2005] Baldan, P., Corradini, A., Esparza, J., Heindel, T., König, B., and Kozioura, V. (2005). Verifying red-black trees. In *Proceedings of COSMICAH '05*. Proceedings available as report RR-05-04 (Queen Mary, University of London).

[BALDAN ET AL., 2006A] Baldan, P., Corradini, A., Heindel, T., König, B., and Sobocinski, P. (2006a). Processes for adhesive rewriting systems. In [Aceto and Ingólfsdóttir, 2006], pages 202–216.

[BALDAN ET AL., 2009] Baldan, P., Corradini, A., Heindel, T., König, B., and Sobociński, P. (2009). Unfolding grammars in adhesive categories. In

Proceedings of CALCO 2009 (Algebra and Coalgebra in Computer Science). Springer. To appear.

[BALDAN ET AL., 2008C] Baldan, P., Corradini, A., and König, B. (2008c). A framework for the verification of infinite-state graph transformation systems. *Information and Computation*, 206:869–907.

[BALDAN ET AL., 2008D] Baldan, P., Corradini, A., and König, B. (2008d). Unfolding graph transformation systems: Theory and applications to verification. In Degano, P., Nicola, R. D., and Meseguer, J., editors, *Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*, pages 16–36. Springer. Lecture notes in computer science 5065.

[BALDAN ET AL., 2008E] Baldan, P., Corradini, A., König, B., and Schwoon, S. (2008e). McMillan's complete prefix for contextual nets. *Transactions on Petri Nets and Other Models of Concurrency (TOPNOC)*, 5100:199–220.

[BALDAN ET AL., 1998A] Baldan, P., Corradini, A., and Montanari, U. (1998a). Concatenable Graph Processes: Relating Processes and Derivation Traces. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, pages 283–295. Springer.

[BALDAN ET AL., 1998B] Baldan, P., Corradini, A., and Montanari, U. (1998b). Unfolding of double-pushout graph grammars is a coreflection. In [Ehrig et al., 2000], pages 145–163.

[BALDAN ET AL., 2007] Baldan, P., Corradini, A., Montanari, U., and Ribeiro, L. (2007). Unfolding semantics of graph transformation. *Information and Computation*, 205(5):733–782.

[BALDAN ET AL., 2006B] Baldan, P., Haar, S., and König, B. (2006b). Distributed unfolding of Petri nets. In *Proceedings of FOSSACS'06*, pages 126–141. Springer. Lecture notes in computer science 3921.

[BARR, 1987] Barr, M. (1987). On categories with effective unions. *Categorical algebra and its applications, Louvain-La-Neuve*, 1348:19–35.

[BERRY AND BOUDOL, 1992] Berry, G. and Boudol, G. (1992). The chemical abstract machine. *Theoretical Computer Science*, 96(1):217–248.

[BONCHI ET AL., 2008] Bonchi, F., Gadducci, F., and Heindel, T. (2008). Parallel and sequential independence for borrowed contexts. In [Ehrig et al., 2008], pages 226–241.

[BONCHI AND HEINDEL, 2006] Bonchi, F. and Heindel, T. (2006). Adhesive DPO parallelism for monic matches. In *Graph Transformation for Verification and Concurrency, GT-VC* 2006.

[BROWN AND JANELIDZE, 1997] Brown, R. and Janelidze, G. (1997). Van Kampen theorems for categories of covering morphisms in lextensive categories. *Journal of Pure and Applied Algebra*, 119(3):255–263.

[BRUNI AND GADDUCCI, 2003] Bruni, R. and Gadducci, F. (2003). Some algebraic laws for spans (and their connections with multirelations). *Electronic Notes in Theoretical Computer Science*, 44(3):175–193.

[BRUNI ET AL., 2001] Bruni, R., Meseguer, J., Montanari, U., and Sassone, V. (2001). Functorial models for Petri nets. *Information and Computation*, 170(2):207–236.

[CARBONI ET AL., 1993] Carboni, A., Lack, S., and Walters, W. (1993). Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra*, 84:145–158.

[CLEMENTINO ET AL., 1996] Clementino, M. M., Giuli, E., and Tholen, W. (1996). Topology in a category: compactness. *Portugaliæ Mathematica*, 53:397–434.

[COCKETT AND GUO, 2007] Cockett, R. and Guo, X. (2007). Join restriction categories and the importance of being adhesive. Unpublished manuscript, slides from CT'07 presentation available at `http://pages.cpsc.ucalgary.ca/~robin/talks/jrCat.pdf`.

[CORRADINI ET AL., 2006] Corradini, A., Heindel, T., Hermann, F., and König, B. (2006). Sesqui-pushout rewriting. *Lecture Notes in Computer Science*, 4178:30–.

[CORRADINI ET AL., 2008] Corradini, A., Hermann, F., and Sobociński, P. (2008). Subobject Transformation Systems. *Applied Categorical Structures*, 16(3):389–419.

[CORRADINI ET AL., 1997] Corradini, A., Montanari, U., Rossi, F., Ehrig, H., Heckel, R., and Löwe, M. (1997). Algebraic approaches to graph transformation – part I: Basic concepts and double pushout approach. In [Rozenberg, 1997], pages 163–246.

[DEGANO ET AL., 1996] Degano, P., Meseguer, J., and Montanari, U. (1996). Axiomatizing the algebra of net computations and processes. *Acta Informaticae*, 33(7):641–667.

[DIEKERT AND MÉTIVIER, 1997] Diekert, V. and Métivier, Y. (1997). *Handbook of formal languages, vol. 3: beyond words*, chapter Partial commutation and traces, pages 457–533. Springer.

[DYCKHOFF AND THOLEN, 1987] Dyckhoff, R. and Tholen, W. (1987). Exponentiable morphisms, partial products and pullback complements. *Journal of Pure and Applied Algebra*, 49(1–2):103–116.

[EHRIG ET AL., 2006] Ehrig, H., Ehrig, K., Prange, U., and Taentzer, G. (2006). *Fundamentals of Algebraic Graph Transformation*. Springer.

[EHRIG ET AL., 1999] Ehrig, H., Engels, G., Kreowski, H.-J., and Rozenberg, G., editors (1999). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 2: Applications, Languages and Tools*. World Scientific.

[EHRIG ET AL., 2000] Ehrig, H., Engels, G., Kreowski, H.-J., and Rozenberg, G., editors (2000). *Theory and Application of Graph Transformations, 6th International Workshop, TAGT'98, Paderborn, Germany, November 16-20, 1998, Selected Papers*, volume 1764 of *Lecture Notes in Computer Science*. Springer.

[EHRIG ET AL., 2004A] Ehrig, H., Engels, G., Parisi-Presicce, F., and Rozenberg, G., editors (2004a). *Graph Transformations, Second International Conference, ICGT 2004, Rome, Italy, September 28 – October 2, 2004, Proceedings*, volume 3256 of *Lecture Notes in Computer Science*. Springer.

[EHRIG ET AL., 1990] Ehrig, H., Habel, A., Kreowski, H.-J., and Parisi-Presicce, F. (1990). From graph grammars to high level replacement systems. In [Ehrig et al., 1991], pages 269–291.

[EHRIG ET AL., 2004B] Ehrig, H., Habel, A., Padberg, J., and Prange, U. (2004b). Adhesive high-level replacement categories and systems. In [Ehrig et al., 2004a], pages 144–160.

[EHRIG ET AL., 1997] Ehrig, H., Heckel, R., Korff, M., Löwe, M., Ribeiro, L., Wagner, A., and Corradini, A. (1997). Algebraic approaches to graph transformation – part II: Single pushout approach and comparison with double pushout approach. In [Rozenberg, 1997], pages 247–312.

[EHRIG ET AL., 2008] Ehrig, H., Heckel, R., Rozenberg, G., and Taentzer, G., editors (2008). *Graph Transformations, 4th International Conference, ICGT 2008, Leicester, United Kingdom, September 7-13, 2008. Proceedings*, volume 5214 of *Lecture Notes in Computer Science*. Springer.

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

[EHRIG ET AL., 1991] Ehrig, H., Kreowski, H.-J., and Rozenberg, G., editors (1991). *Graph-Grammars and Their Application to Computer Science, 4th International Workshop, Bremen, Germany, March 5-9, 1990, Proceedings*, volume 532 of *Lecture Notes in Computer Science*. Springer.

[EHRIG AND LÖWE, 1993] Ehrig, H. and Löwe, M. (1993). Categorical principles, techniques and results for high-level-replacement systems in computer science. *Applied Categorical Structures*, 1(1):21–50.

[EHRIG ET AL., 1973] Ehrig, H., Pfender, M., and Schneider, H. J. (1973). Graph-grammars: An algebraic approach. In *14th Annual Symposium on Switching and Automata Theory*, pages 167–180. Institute of Electrical and Electronics Engineers.

[EHRIG AND PRANGE, 2006] Ehrig, H. and Prange, U. (2006). Weak adhesive high-level replacement categories and systems: A unifying framework for graph and Petri net transformations. In Futatsugi, K., Jouannaud, J.-P., and Meseguer, J., editors, *Essays Dedicated to Joseph A. Goguen*, volume 4060 of *Lecture Notes in Computer Science*, pages 235–251. Springer.

[EHRIG ET AL., 2004C] Ehrig, H., Prange, U., and Taentzer, G. (2004c). Fundamental theory for typed attributed graph transformation. In [Ehrig et al., 2004a], pages 161–177.

[ESPARZA AND HELJANKO, 2008] Esparza, J. and Heljanko, K., editors (2008). *Unfoldings: A Partial-Order Approach to Model Checking*. Springer.

[FIADEIRO, 2005] Fiadeiro, J. L. (2005). *Categories For Software Engineering*. Springer.

[FREYD, 1972] Freyd, P. (1972). Aspects of topoi. *Bulletin of the Australian Mathematical Society*, 7:1–76.

[GOLTZ AND REISIG, 1983] Goltz, U. and Reisig, W. (1983). The non-sequential behavior of Petri nets. *Information and Control*, 57(2/3):125–147.

[HABEL ET AL., 2001] Habel, A., Müller, J., and Plump, D. (2001). Double-pushout graph transformation revisited. *Mathematical Structures in Computer Science*, 11(5):637–688.

[HAYMAN AND WINSKEL, 2008] Hayman, J. and Winskel, G. (2008). The unfolding of general Petri nets. In Hariharan, R., Mukund, M., and Vinay, V., editors, *FSTTCS*, volume 08004 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

[HECKEL AND WAGNER, 1995] Heckel, R. and Wagner, A. (1995). Ensuring consistency of conditional graph grammars – a constructive approach. *Electronic Notes in Theoretical Computer Science*, 2:118–126.

[HEINDEL, 2008] Heindel, T. (2008). Grammar morphisms and weakly adhesive categories. In [Ehrig et al., 2008], pages 493–495.

[HEINDEL, 2009] Heindel, T. (2009). Towards secrecy for rewriting in weakly adhesive categories. *Electronic Notes in Theoretical Computer Science*, 229(3):97–115.

[HEINDEL AND SOBOCIŃSKI, 2009] Heindel, T. and Sobociński, P. (2009). Van Kampen colimits as bicolimits in Span. In *Proceedings of CALCO 2009 (Algebra and Coalgebra in Computer Science)*. Springer. To appear.

[HILDEBRANDT AND SASSONE, 1996] Hildebrandt, T. and Sassone, V. (1996). Transition systems with independence and multi-arcs. In Holzmann, G. J., Peled, D. A., and Pratt, V. R., editors, *Proceedings of POMIV'96*, volume 29 of *Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society.

[JANELIDZE ET AL., 2004] Janelidze, G., Sobral, M., and Tholen, W. (2004). Beyond Barr exactness: Effective descent morphisms. In Pedicchio, M. C. and Tholen, W., editors, *Categorical Foundations: Special Topics in Order, Topology, Algebra, and Sheaf Theory*. Cambridge University Press.

[JOHNSTONE, 1977] Johnstone, P. T. (1977). *Topos theory*. Academic Press.

[JOHNSTONE, 2002] Johnstone, P. T. (2002). *Sketches of an elephant: a topos theory compendium*. Oxford University Press.

[JOHNSTONE ET AL., 2007] Johnstone, P. T., Lack, S., and Sobociński, P. (2007). Quasitoposes, quasiadhesive categories and artin glueing. In *Algebra and Coalgebra in Computer Science, Calco 2007*, volume 4626 of *Lecture Notes in Computer Science*. Springer.

[JOYAL ET AL., 1996] Joyal, A., Nielsen, M., and Winskel, G. (1996). Bisimulation from open maps. *Information and Computation*, 127(2):164–185.

[KENNAWAY, 1990] Kennaway, R. (1990). Graph rewriting in some categories of partial morphisms. In [Ehrig et al., 1991], pages 490–504.

[KÖNIG AND KOZIOURA, 2006A] König, B. and Kozioura, V. (2006a). Augur 2—a new version of a tool for the analysis of graph transformation systems. In *Proceedings of GT-VMT '06 (Workshop on Graph Transformation and Visual Modeling Techniques)*, pages 63–72. ENTCS Vol. 175, N 4.

[KÖNIG AND KOZIOURA, 2006B] König, B. and Kozioura, V. (2006b). Counterexample-guided abstraction refinement for the analysis of graph transformation systems. In *Proceedings of TACAS'06*, pages 154–169. Springer. Lecture notes in computer science 1855.

[KREOWSKI, 1977] Kreowski, H.-J. (1977). *Manipulation von Graph Transformationen*. PhD thesis, Technische Universitat Berlin.

[KREOWSKI, 1986] Kreowski, H.-J. (1986). Is parallelism already concurrency? part 1: Derivations in graph grammars. In Ehrig, H., Nagl, M., Rozenberg, G., and Rosenfeld, A., editors, *Graph-Grammars and Their Application to Computer Science*, volume 291 of *Lecture Notes in Computer Science*, pages 343–360. Springer.

[LACK AND SOBOCIŃSKI, 2004] Lack, S. and Sobociński, P. (2004). Adhesive categories. In Walukiewicz, I., editor, *FOSSACS*, volume 2987 of *Lecture Notes in Computer Science*, pages 273–288. Springer.

[LACK AND SOBOCIŃSKI, 2005] Lack, S. and Sobociński, P. (2005). Adhesive and quasiadhesive categories. *Theoretical Informatics and Applications*, 39(2):511–546.

[LACK AND SOBOCIŃSKI, 2006] Lack, S. and Sobociński, P. (2006). Toposes are adhesive. In Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., and Rozenberg, G., editors, *ICGT*, volume 4178 of *Lecture Notes in Computer Science*, pages 184–198. Springer.

[LAWVERE, 1963] Lawvere, F. W. (1963). Functorial Semantics of Algebraic Theories. *Proceedings of the National Academy of Sciences*, 50(5):869–872.

[LLABRÉS AND ROSSELLÓ, 1998] Llabrés, M. and Rosselló, F. (1998). Pushout complements for arbitrary partial algebras. In [Ehrig et al., 2000], pages 131–144.

[LÖWE AND EHRIG, 1990] Löwe, M. and Ehrig, H. (1990). Algebraic approach to graph transformation based on single pushout derivations. In Möhring, R. H., editor, *WG '90: Proceedings of the 16th International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 484 of *Lecture Notes in Computer Science*, pages 338–353. Springer.

[MAC LANE, 1998] Mac Lane, S. (1998). *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer.

[MAY, 2003] May, J. P. (2003). Finite topological spaces. Notes for REU (2003).

[MAZURKIEWICZ, 1986] Mazurkiewicz, A. W. (1986). Trace theory. In Brauer, W., Reisig, W., and Rozenberg, G., editors, *Advances in Petri nets*, volume 255 of *Lecture Notes in Computer Science*, pages 279–324. Springer.

[MCMILLAN, 1992] McMillan, K. L. (1992). *Symbolic Model Checking*. PhD thesis, Carnegie Mellon University.

[MCMILLAN, 1995] McMillan, K. L. (1995). A technique of state space search based on unfolding. *Formal Methods in System Design*, 6(1):45–65.

[MESEGUER ET AL., 1997] Meseguer, J., Montanari, U., and Sassone, V. (1997). On the semantics of place/transition Petri nets. *Mathematical Structures in Computer Science*, 7(04):359–397.

[MONSERRAT ET AL., 1997] Monserrat, M., Rosselló, F., Torrens, J., and Valiente, G. (1997). Single-pushout rewriting in categories of spans I: The general setting. Informe d'investigació, Department of Software (LSI) Universitat Politècnica de Catalunya.

[MONTANARI ET AL., 1996] Montanari, U., Corradini, A., and Rossi, F. (1996). Graph processes. *Fundamenta Informaticae*, 26:241–265.

[NIEFIELD, 1982] Niefield, S. B. (1982). Cartesianness: topological spaces, uniform spaces, and affine schemes. *Journal Pure Applied Algebra*, 23:147–167.

[NIELSEN ET AL., 1981] Nielsen, M., Plotkin, G. D., and Winskel, G. (1981). Petri nets, event structures and domains, part I. *Theoretical Computer Science*, 13:85–108.

[PADBERG, 1993] Padberg, J. (1993). Survey of high-level replacement systems. Technical report, Technische Universität Berlin.

[PENON, 1977] Penon, J. (1977). Sur les quasi-topos. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 18(2):181–218.

[PETRI, 1962] Petri, C. A. (1962). *Kommunikation mit Automaten.* PhD thesis, Bonn: Institut für Instrumentelle Mathematik.

[PIERCE, 1991] Pierce, B. C. (1991). *Basic Category Theory for Computer Scientists.* MIT Press.

[PLUMP, 1997] Plump, D. (1997). Term graph rewriting. In [Ehrig et al., 1999], pages 247–312.

[PRANGE, 2007] Prange, U. (2007). Algebraic High-Level Nets as Weak Adhesive HLR Categories. *Electronic Communications of the EASST*, 2:1–13.

[PRATT, 1982] Pratt, V. R. (1982). On the composition of processes. In *Proceedings of the 9th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 213–223. ACM New York, NY, USA.

[RAOULT, 1984] Raoult, J.-C. (1984). On graph rewritings. *Theoretical Computer Science*, 32:1–24.

[RIBEIRO, 1996] Ribeiro, L. (1996). *Parallel composition and unfolding semantics of graph grammars*. PhD thesis, Technische Universität Berlin.

[ROBINSON AND ROSOLINI, 1988] Robinson, E. and Rosolini, G. (1988). Categories of partial maps. *Information and Computation*, 79(2):95–130.

[ROSOLINI, 1986] Rosolini, G. (1986). *Continuity and Effectiveness in Topoi*. PhD thesis, University of Oxford.

[ROZENBERG, 1997] Rozenberg, G., editor (1997). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific.

[SASSONE, 1996] Sassone, V. (1996). An axiomatization of the algebra of Petri net concatenable processes. *Theoretical Computer Science*, 170(1-2):277–296.

[SASSONE ET AL., 1996] Sassone, V., Nielsen, M., and Winskel, G. (1996). Models for concurrency: Towards a classification. *Theoretical Computer Science*, 170(1-2):297–348.

[VAN DEN BROEK, 1991] van den Broek, P. M. (1991). Algebraic graph rewriting using a single pushout. In Abramsky, S. and Maibaum, T. S. E., editors, *TAPSOFT, Vol.1*, volume 493 of *Lecture Notes in Computer Science*, pages 90–102. Springer.

[VON BERTALANFFY, 1973] von Bertalanffy, L. (1973). *General System theory: Foundations, Development, Applications*. Penguin Books.

[WINSKEL, 1985] Winskel, G. (1985). Petri nets, morphisms and compositionality. *Lecture Notes In Computer Science; Vol. 222*, pages 453–477.

[WYLER, 1991] Wyler, O. (1991). *Lecture notes on topoi and quasitopoi*. World Scientific.

Appendix

Preliminaries from Category theory

A

This section briefly introduces the necessary notions from category theory. Thus it is rather an enumeration of definitions, propositions and facts than a detailed introduction to the topic. The book [Pierce, 1991] is a more verbose basic introduction to category theory; the standard reference is [Mac Lane, 1998]. As it is common practice, issues of size which are connected with formal notions like sets and classes are often not elaborated on and hence often the word 'collection' is used (instead of 'set' or 'class').

DEFINITION A.1 (Graphs and categories) A *graph* H is given by a quadruple $H = \langle V_H, E_H, s_H, t_H \rangle$ where

- V_H is a collection of *vertices* or *nodes*,
- E_H is a collection of *edges* or *arcs*,
- and $s_H \colon E_H \to V_H$ and $t_H \colon E_H \to V_H$ are operators mapping each edge to its *source* and *target*, respectively.

Any of the expressions '$e \colon u \to v$ *in* H', '$e \colon v \leftarrow u$ *in* H', '$u \mathrel{-e\!\!\rightarrow} v$ *in* H' and '$v \mathrel{\leftarrow\!\!e-} u$ *in* H' is used for saying that $e \in E_H$ is an edge with source $u = s_H(e)$ and target $v = t_H(e)$. Further the *collection of composable*[80] *pairs* $E_H \times_{V_H} E_H$ *in* H is $E_H \times_{V_H} E_H = \{\langle e', e \rangle \mid e, e' \in E_H \text{ and } s_H(e') = t_H(e)\}$, which contains all pairs $u \mathrel{-e\!\!\rightarrow} v \mathrel{-e'\!\!\rightarrow} w$ of "consecutive" edges.

A *category* $\mathbb{C}$ is determined by the following data and axioms.

DATA The data of a category $\mathbb{C}$ are given by a triple $\langle H, \mathrm{id}^{\mathbb{C}}, \circ_{\mathbb{C}} \rangle$ where

- $H = \langle V_H, E_H, s_H, t_H \rangle$ is a graph,
- $\mathrm{id}^{\mathbb{C}}_{\_} \colon V_H \to E_H$ is an operation which maps each $A \in V_H$ to the *identity* $\mathrm{id}^{\mathbb{C}}_A \colon A \leftarrow A$ *on* A,
- $\_ \circ_{\mathbb{C}} \_ \colon E_H \times_{V_H} E_H \to E_H$ is a binary operation which takes any composable pair $\langle f \colon C \leftarrow B, g \colon B \leftarrow A \rangle$ in H as input and produces its *composition* $f \circ_{\mathbb{C}} g \colon C \leftarrow A$.

The graph H is called the *underlying graph* of $\mathbb{C}$ and denoted by $\mathcal{U}\mathbb{C}$. The elements of V_H are called *objects*, and the elements of E_H are referred to

[80] This term is introduced for graphs since it is not only used for the definition of the data of a category but also for the free category of a graph (cf. Example A.4).

as *morphisms* or *arrows* of the category $\mathbb{C}$. Moreover, it is common to write '$\mathrm{ob}(\mathbb{C})$' and '$\mathrm{ar}(\mathbb{C})$' instead of 'V_H' and 'E_H', respectively.

The expression '$A \in \mathbb{C}$' is used to indicate that $A \in \mathrm{ob}(\mathbb{C}) = V_H$ is an object of $\mathbb{C}$. Similarly '$f\colon A \to B$ in $\mathbb{C}$' expresses that $f\colon A \to B$ in H; in this situation A and B are referred to as the *domain* and *codomain* of f, respectively, written $A = \mathrm{dm}\, f$ and $B = \mathrm{cd}\, f$. Finally, the collection of all morphisms with domain A and codomain B is the *homset* $\mathbb{C}(A, B) = \{f \in \mathrm{ar}(\mathbb{C}) \mid f\colon A \to B \text{ in } \mathbb{C}\}$.

AXIOMS The data of a category $\mathbb{C}$ must satisfy the following axioms.

(i) For all of objects $A, B \in \mathbb{C}$ and all arrows $f\colon B \leftarrow A$ and $g\colon A \leftarrow B$, the identity id_A must satisfy

$$f = f \circ \mathrm{id}_A \qquad \text{and} \qquad \mathrm{id}_A \circ g = g, \tag{3}$$

i.e. identities are left and right cancellable.

(ii) For all of objects $A, B, C, D \in \mathbb{C}$ and morphisms $h\colon D \leftarrow C$, $g\colon C \leftarrow B$, $f\colon B \leftarrow A$ in $\mathbb{C}$, the composition operation $\_ \circ \_$ must satisfy

$$h \circ (g \circ f) = (h \circ g) \circ f, \tag{4}$$

i.e. composition is associative.

EXAMPLE A.2 (The categories of $\mathbb{S}\mathrm{ets}$ and $\mathbb{f}\mathbb{S}\mathrm{ets}$) The category of $\mathbb{S}\mathrm{ets}$ has the class of all sets as objects and its arrows are all functions between any two sets. The identity on a set $M \in \mathbb{S}\mathrm{ets}$ is the identity function, i.e. the function described by $\mathrm{id}_M(m) = m$ for all $m \in M$; composition of two arrows $f\colon M \to N$ and $g\colon N \to K$ in $\mathbb{S}\mathrm{ets}$ is given by function composition $g \circ f$, which is defined by $(g \circ f)(m) = g(f(m))$ for all $m \in M$. In a similar way one obtains the *small* category of $\mathbb{f}\mathbb{S}\mathrm{ets}$, which has the *set* of finite sets as the collection of objects.

EXAMPLE A.3 (Monoids are one object categories) Each monoid $\langle M, e, \cdot \rangle$ (where M is the set of all elements of the monoid, $e \in M$ is the unit, and $\_ \cdot \_\colon M \times M \to M$ is the multiplication) is essentially a category with one object $\star$ and a collection of arrows that contains for each element $a \in M$ the arrow $a\colon \star \to \star$. The identity on $\star$ is given by $e =: \mathrm{id}_\star\colon \star \to \star$ and composition of two arrows $a, b \in M$ is nothing else but multiplication, in other words

$a \circ b := a \cdot b$. Conversely, each category $\mathbb{C}$ that has only one object $X \in \mathbb{C}$ is essentially a monoid, namely $\langle \mathbb{C}(X,X), \mathrm{id}_X^{\mathbb{C}}, \circ_{\mathbb{C}} \rangle$; indeed, the latter construction applies to any object X in any category $\mathbb{C}$.[81]

Example A.4 (Free categories of graphs) This example demonstrates how each graph $G = \langle V_G, E_G, s_G \colon E_G \to V_G, t_G \colon E_G \to V_G \rangle$ gives rise to a *free category* in a similar way as a set Σ can be extended to a free monoid Σ^*. The free category of G has V_G as the set of objects. Morphisms are sequences of composable edges, i.e. for a pair of objects $A, B \in V_G$ a morphism from A to B is a finite sequence of edges $\langle e_0, \ldots, e_n \rangle \in E_G^*$ such that $\langle e_i, e_{i+1} \rangle \in E_H \times_{V_H} E_H$ is a composable pair for each $i \in \{0, \ldots, n-1\}$. The identity on A is given by the empty sequence, and composition of morphisms is achieved by multiplication in the free monoid E_G^*. In this sense, each graph is (a presentation of) a category.

Another, substantially different way to obtain a category involving graphs is the congregation of the collections of all graphs into a category which is based on the notion of graph morphism; moreover, graph morphisms will be the central concept in the definition of structure preserving maps between categories, namely functors.

Definition A.5 (Graph morphisms) Let K and L be two graphs, where $K = \langle V_K, E_K, s_K, t_K \rangle$ and $L = \langle V_L, E_L, s_L, t_L \rangle$. A *graph morphism φ from K to L* is a pair $\varphi = \langle \varphi_V, \varphi_E \rangle$ where $\varphi_V \colon V_K \to V_L$ and $\varphi_E \colon E_K \to E_L$ are mappings such that the following two equations hold.

$$s_L \circ \varphi_E = \varphi_V \circ s_K \qquad\qquad t_L \circ \varphi_E = \varphi_V \circ t_K. \tag{5}$$

Example A.6 (The category of $\mathtt{Graphs}$) For this, take the class of all (small) graphs as the collection of objects of the category $\mathtt{Graphs}$. Morphisms between two objects $K, L \in \mathtt{Graphs}$, i.e. between two graphs, are all graph morphism $\varphi \colon K \to L$; identities and composition are defined componentwise on nodes and edges, i.e. $\mathrm{id}_K = \langle \mathrm{id}_{V_K}, \mathrm{id}_{E_K} \rangle \colon K \to K$ and $\psi \circ \varphi = \langle \psi_V \circ \varphi_V, \psi_E \circ \varphi_E \rangle$ where K is graph and $\varphi \colon K \to L$ and $\psi \colon L \to M$ are graph morphisms.

Further examples of categories can be obtained by constructing new categories from old ones, which is in analogy to the fact that for any two sets A, B there exist their disjoint union $A + B$, their cartesian product $A \times B$ and the function space B^A, which contains all functions from A to B. In fact,

[81] To be precise, $\mathbb{C}$ must be assumed to be locally small, i.e. the homset $\mathbb{C}(X,X)$ is a proper set.

the latter three constructions have exact counterparts for categories. Another, fundamentally different operation associates to each category its *opposite*, which is obtained by reversing all arrows.

☼ Definition A.7 (Opposite category) Let $\mathbb{C}$ be a category. Its opposite category $\mathbb{C}^{\mathrm{op}}$, has the same objects and arrows as $\mathbb{C}$, i.e. $\mathrm{ob}(\mathbb{C}^{\mathrm{op}}) = \mathrm{ob}(\mathbb{C})$, and $\mathrm{ar}(\mathbb{C}^{\mathrm{op}}) = \mathrm{ob}(\mathbb{C})$. However, in $\mathbb{C}^{\mathrm{op}}$, arrows are oriented in the opposite direction, i.e. $f\colon A \to B$ in $\mathbb{C}^{\mathrm{op}}$ if and only if $f\colon B \to A$ in $\mathbb{C}$. The identity on A in $\mathbb{C}^{\mathrm{op}}$ is the same as in $\mathbb{C}$, i.e. $\mathrm{id}_A^{\mathbb{C}^{\mathrm{op}}} = \mathrm{id}_A^{\mathbb{C}}$, and also composition is inherited from $\mathbb{C}$, i.e. for all morphisms $f\colon A \to B$ and $g\colon B \to C$ in $\mathbb{C}^{\mathrm{op}}$ their composite is defined by $g \circ_{\mathbb{C}^{\mathrm{op}}} f = f \circ_{\mathbb{C}} g$. ☼

Issues of size, distinctions between sets and proper classes, and possible ways to avoid connected problems are usually not elaborated on in this thesis. Nevertheless, at this point it might be worth to mention that the underlying graph $\mathcal{U}\mathbb{G}\mathtt{raphs}$, which has small graphs as nodes and graph morphisms as edges, is not small, i.e. it has a proper class as the collection of nodes, which consists of all small graphs. Similarly, also $\mathcal{U}\mathbb{S}\mathtt{ets}$ is a *large* graph.

☼ Definition A.8 (Diagrams) Let J be a graph and let $\mathbb{C} = \langle H, \mathrm{id}, \circ \rangle$ be a category. Then a *diagram* $\mathcal{D}$ *of shape* J *in* $\mathbb{C}$ is a graph morphism $\mathcal{D}\colon J \to H$. Such a diagram $\mathcal{D}$ is *commuting* or *commutes* if for any pair of nodes $u, u' \in V_J$ and any pair of paths $u \mathrel{-}e_1\!\!\to v_1 \cdots v_n \mathrel{-}e_n\!\!\to u'$, $u \mathrel{-}d_1\!\!\to w_1 \cdots w_m \mathrel{-}d_m\!\!\to u'$ from u to u' in J, the following equation holds in $\mathbb{C}$.

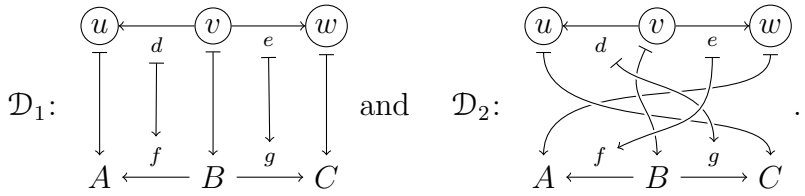
$$\mathcal{D}_E(e_n) \circ \cdots \circ \mathcal{D}_E(e_1) = \mathcal{D}_E(d_m) \circ \cdots \circ \mathcal{D}_E(d_1) \tag{6}$$

The fact that $\mathcal{D}$ is a diagram is expressed by writing '$\mathcal{D}\colon J \to \mathbb{C}$', where J is the shape of $\mathcal{D}$. ☼

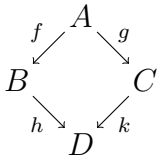
Often one writes $f \fatsemi g$ for $g \circ f$. For example Equation (6) could be written as $\mathcal{D}_E(e_1) \fatsemi \cdots \fatsemi \mathcal{D}_E(e_n) = \mathcal{D}_E(d_1) \fatsemi \cdots \fatsemi \mathcal{D}_E(d_m)$. This is to the effect that in the latter equation the edges are listed in the order in which they occur on the paths through the diagram $\mathcal{D}$ – hence $\_ \fatsemi \_$ is called the *diagrammatic* notation for composition.

☙ Example A.9 (Frequently used Diagrams) Let $\mathbb{C}$ be a category. Then a $\mathbb{C}$-*span* is a diagram of shape $\textcircled{u} \leftarrow d - \textcircled{v} - e \rightarrow \textcircled{w}$, a $\mathbb{C}$-*cospan* is a diagram of shape $\textcircled{u} - d \rightarrow \textcircled{v} \leftarrow e - \textcircled{w}$, and a diagram of shape $\textcircled{x} \leftarrow e' - \textcircled{u} \leftarrow d - \textcircled{v} - e \rightarrow \textcircled{w} - d' \rightarrow \textcircled{x}$ is called a $\mathbb{C}$-square. As the "names" of the nodes and edges of the involved graphs are usually irrelevant, one can present a diagram using its "image" in the category $\mathbb{C}$. For example a $\mathbb{C}$-span is given by just writing '$A \leftarrow f - B - g \rightarrow C$' where $A, B, C \in \mathbb{C}$ and $f\colon B \to A$ and $g\colon B \to C$ in $\mathbb{C}$. However this practice

relies on the left-right distinction since there are actually two ways in which the "image" $A \leftarrow f - B - g \rightarrow C$ arises from a diagram of shape $\textcircled{u} \leftarrow d - \textcircled{v} - e \rightarrow \textcircled{w}$, namely



The intended meaning of '$A \leftarrow f - B - g \rightarrow C$' is however $\mathcal{D}_1$ as it "preserves" left and right. Relying on this convention, the expression '$A - h \rightarrow B \leftarrow k - C$' singles out exactly one $\mathbb{C}$-cospan.



Finally a *square* is given by a graphic as displayed; such a square commutes if and only if the equation $g \fatsemi k = f \fatsemi h$ holds. Again, given the graph $\textcircled{x} \leftarrow e' - \textcircled{u} \leftarrow d - \textcircled{v} - e \rightarrow \textcircled{w} - d' \rightarrow \textcircled{x}$, there are actually two ways in which the displayed graphic arises as a diagram of it: the node u can be mapped either to B or to C. ❦

Usually the ambiguities that are discussed in the preceding example do not cause problems. However, for the definition of transformations of diagrams, or *cones* and *cocones* for a diagram, one requires complete information about the involved diagrams.

❦ Example A.10 (Categories of diagrams) Let $J = \langle V_J, E_J, s_J, t_J \rangle$ be a graph and let $\mathbb{C} = \langle H, \mathrm{id}, \circ \rangle$ be a category. Then the J-shaped diagrams form a category $\mathbb{Diag}[J, \mathbb{C}]$; its collection of objects contains all J-shaped diagrams and a morphism $\mathbb{f}$ from a diagram $\mathcal{D} \colon J \to \mathbb{C}$ to a diagram $\mathcal{E} \colon J \to \mathbb{C}$ is a family of $\mathbb{C}$-morphisms $\mathbb{f} = \{f_u \colon \mathcal{D}_V(u) \to \mathcal{E}_V(u)\}_{u \in V_J}$ such that the equation $\mathcal{E}_E(e) \circ f_u = f_v \circ \mathcal{D}_E(e)$ holds true for every edge $e \colon u \to v$ in J. ❦

☼ Definition A.11 (Cones and cocones, limits and colimits) Let $\mathbb{C}$ be a category, J a graph and $\mathcal{D} \colon J \to \mathbb{C}$ a $\mathbb{C}$-diagram of shape J. Then a *cone* $\mathbb{f}$ *for* $\mathcal{D}$ or a $\mathcal{D}$*-cone* is a family of $\mathbb{C}$-morphisms $\mathbb{f} = \{f_u \colon C \to \mathcal{D}_V(u)\}_{u \in V_J}$ such that $f_u \fatsemi \mathcal{D}_E(e) = f_v$ holds for all edges $e \colon u \to v$ in J; the object $C \in \mathbb{C}$ is the *cone object* of $\mathbb{f}$. In such a situation one writes $\mathbb{f} \colon \Delta_C^J \to \mathcal{D}$ or just $\mathbb{f} \colon C \to \mathcal{D}$ if the latter does result in ambiguities.

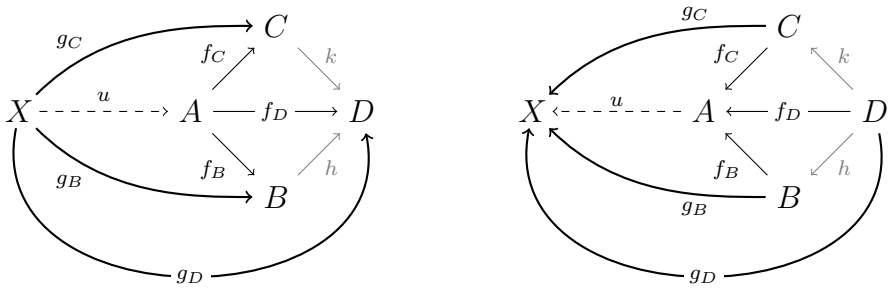
Further a *limit* $\mathbb{l}$ *for* $\mathcal{D}$ is a cone $\mathbb{l} = \{l_u \colon L \to \mathcal{D}_V(u)\}_{u \in V_J}$ for $\mathcal{D}$ such that for any other $\mathcal{D}$-cone $\mathbb{f} = \{f_u \colon C \to \mathcal{D}_V(u)\}_{u \in V_J}$ there exists a unique morphism $m \colon C \to L$ such that $f_u = m \circ l_u$ holds for all $u \in V_J$.

A *cocone* $\mathbb{g}$ *for* $\mathcal{D}$ or a $\mathcal{D}$*-cocone* is a family $\mathbb{g} = \{g_u \colon \mathcal{D}_V(u) \to C\}_{u \in V_J}$ such that $g_u = \mathcal{D}_E(e) \fatsemi g_v$ holds for all edges $e \colon u \to v$ in J; the object $C \in \mathbb{C}$ is the *cocone object* of $\mathbb{g}$. This can also be expressed by writing $\mathbb{g} \colon \mathcal{D} \to \Delta_C^J$.

Finally a *colimit* $\mathbb{t}$ *for* $\mathcal{D}$ is a cocone $\mathbb{t} = \{t_u \colon \mathcal{D}_V(u) \to T\}_{u \in V_J}$ for $\mathcal{D}$ such that for any other $\mathcal{D}$-cocone $\mathbb{g} = \{g_u \colon \mathcal{D}_V(u) \to C\}_{u \in V_J}$ there exists a unique morphism $n \colon T \to C$ such that $g_u = t_u \fatsemi n$ holds for all $u \in V_J$. ☼

As a basic fact, limits and colimits are determined uniquely up to a unique isomorphism, i.e. given two colimits $\mathbb{t} \colon \mathcal{D} \to \Delta_T^J$ and $\mathbb{s} \colon \mathcal{D} \to \Delta_S^J$ there is a unique isomorphism $i \colon T \to S$ such that $s_u = t_u \fatsemi i$ holds for all $u \in V_J$. Moreover, a colimit in $\mathbb{C}$ is essentially the same as a limit in $\mathbb{C}^{\mathrm{op}}$.

☙ Example A.12 (Frequently used limits and colimits) A limit for a cospan $B \xrightarrow{h} D \xleftarrow{k} C$ is given by a triple of arrows $\langle f_B, f_D, f_C \rangle$ such that both $f_B \fatsemi h = f_D$ and $f_D = f_C \fatsemi k$ hold, and moreover for any other triple $\langle g_B, g_D, g_C \rangle$ satisfying $g_B \fatsemi h = g_D = g_C \fatsemi k$, there is a uniquely determined arrow u for which the three equations $u \fatsemi f_B = g_B$, $u \fatsemi f_D = g_D$, and $u \fatsemi f_C = g_C$ hold. How the property which makes $\langle f_B, f_D, f_C \rangle$ a limit for $B \xrightarrow{h} D \xleftarrow{k} C$ can be described diagrammatically is shown in Figure 85(a). A limit of a cospan is called a *pullback*.



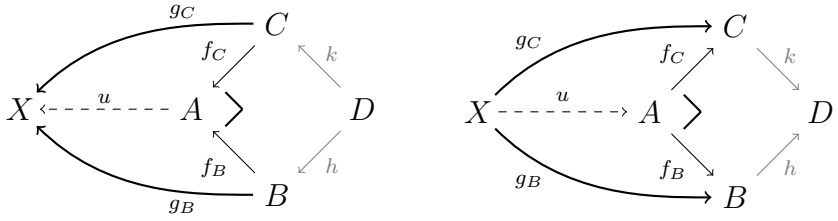
(a) Limit of a cospan

(b) Colimit of a span

Figure 85: Example for limits and colimits

Dually a colimit for a span $B \xleftarrow{h} D \xrightarrow{k} C$ is given by a triple of arrows $\langle f_B, f_D, f_C \rangle$ such that both $f_B \circ h = f_D$ and $f_D = f_C \circ k$ hold, and moreover for any other triple $\langle g_B, g_D, g_C \rangle$ satisfying $g_B \circ h = g_D = g_C \circ k$, there is a uniquely determined arrow u for which the three equations $u \circ f_B = g_B$, $u \circ f_D = g_D$, and $u \circ f_C = g_C$ hold. This situation can be depicted as shown in Figure 85(b). A colimit of a span is a *pushout*. ❧

In the case of pushouts, the "diagonal" arrow can be omitted, i.e. given a pushout $\langle f_B, f_D, f_C \rangle$ of a span $B \xleftarrow{h} D \xrightarrow{k} C$ the morphism f_D can be reconstructed from either f_B or f_C using the equations $f_D = f_B \circ h$ and



(a) Pushout square (b) Pullback square

Figure 86: Simplified specification of pushouts and pullbacks

$f_D = f_C \circ k$; and similarly it is not necessary to mention the "diagonal" of pullbacks. Hence it has become common to describe pushouts and pullbacks in the more succinct way that is illustrated in Figure 86. In this way one obtains the following more common definition of pushouts and pullbacks.

Definition A.13 (Pushouts and pullbacks) Let $B \leftarrow_h - D -_k \rightarrow C$ be a span. A *pushout* of $B \leftarrow_h - D -_k \rightarrow C$ is a co-span $B -_{f_B} \rightarrow A \leftarrow_{f_C} - C$ such that for any other cospan $B -_{g_B} \rightarrow X \leftarrow_{g_C} - C$ there exists a unique arrow $u\colon A \to X$ satisfying the two equations $g_B = u \circ f_B$ and $g_C = u \circ f_C$ (see also Figure 86(a)). In such a situation, the object D is called the *pushout object*.

Let $B -_h \rightarrow D \leftarrow_k - C$ be a cospan. A *pullback* of $B \leftarrow_h - D -_k \rightarrow C$ is a span $B \leftarrow_{f_B} - A -_{f_C} \rightarrow C$ such that for any other span $B \leftarrow_{g_B} - X -_{g_C} \rightarrow C$ there exists a unique arrow $u\colon X \to A$ satisfying the two equations $g_B = u \fatsemi f_B$ and $g_C = u \fatsemi f_C$ (see also Figure 86(b)). In such a situation, the object D is called the *pullback object*.

A basic, but central Lemma is the so-called Pushout Lemma, and the Pullback Lemma is obtained by reversing the direction of the arrows.

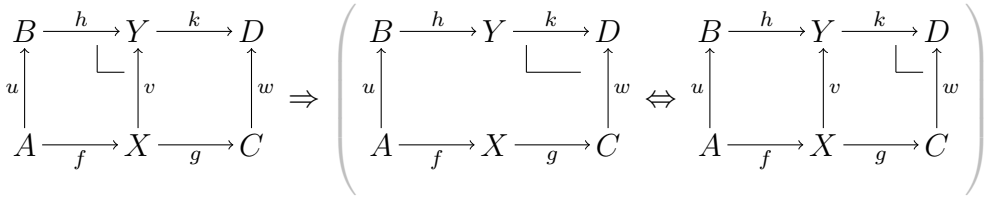


Figure 87: Illustration of the Pushout Lemma

Lemma A.14 (Pushout Lemma) Let $\mathbb{C}$ be any category and consider a pair of neighbouring commuting squares as shown in the left hand diagram in

Figure 87 such that the left square $\begin{smallmatrix} B \\ A \end{smallmatrix}\!\uparrow\!\overset{\rightarrow}{\underset{\rightarrow}{\sqsubset}}\!\uparrow \begin{smallmatrix} Y \\ X \end{smallmatrix}$ is a pushout square, i.e. $B \mathrel{-}h\mathrel{\rightarrow} Y \mathrel{\leftarrow}v\mathrel{-} X$ is a pushout of $B \mathrel{\leftarrow}u\mathrel{-} A \mathrel{-}f\mathrel{\rightarrow} X$.

Then the outer square $\begin{smallmatrix} B \\ A \end{smallmatrix}\!\uparrow\!\overset{\rightarrow}{\underset{\rightarrow}{\sqsubset}}\!\uparrow \begin{smallmatrix} D \\ C \end{smallmatrix}$ shown in the center diagram in Figure 87 is a pushout square, i.e. $B \mathrel{-}h\fatsemi k\mathrel{\rightarrow} D \mathrel{\leftarrow}w\mathrel{-} C$ is a pushout of $B \mathrel{\leftarrow}u\mathrel{-} A \mathrel{-}f\fatsemi g\mathrel{\rightarrow} C$, if and only if the right square $\begin{smallmatrix} Y \\ X \end{smallmatrix}\!\uparrow\!\overset{\rightarrow}{\underset{\rightarrow}{\sqsubset}}\!\uparrow \begin{smallmatrix} D \\ C \end{smallmatrix}$ is a pushout square as depicted in the right hand diagram in Figure 87, i.e. if and only if $Y \mathrel{-}k\mathrel{\rightarrow} D \mathrel{\leftarrow}w\mathrel{-} C$ is a pushout of $Y \mathrel{\leftarrow}v\mathrel{-} X \mathrel{-}g\mathrel{\rightarrow} C$.

A category is said to have pullbacks if for any cospan $B \mathrel{-}h\mathrel{\rightarrow} D \mathrel{\leftarrow}k\mathrel{-} C$ there exists some pullback of $B \mathrel{-}h\mathrel{\rightarrow} D \mathrel{\leftarrow}k\mathrel{-} C$, and similarly, a category has pushouts if for every span $B \mathrel{\leftarrow}h\mathrel{-} D \mathrel{-}k\mathrel{\rightarrow} C$ there exists some pushout of $B \mathrel{\leftarrow}h\mathrel{-} D \mathrel{-}k\mathrel{\rightarrow} C$. The general definition is as follows.

☼ Definition A.15 (Existence of limits and colimits) Let J be a graph, and let $\mathbb{C}$ be a category. Then $\mathbb{C}$ *has colimits (limits) of shape* J if for each diagram $\mathcal{D}\colon J \to \mathcal{U}\mathbb{C}$, some colimit $\mathbb{c}\colon \mathcal{D} \to \Delta_C^J$ (limit $\mathbb{l}\colon \Delta_L^J \to \mathcal{D}$) exists in $\mathbb{C}$. ☼

However, it might happen that the existence of limits and colimits is not only dependent on the shape of the involved diagrams, but also additional conditions must be satisfied. For example, given a category $\mathbb{C}$ and a class of $\mathbb{C}$-arrows $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$, the category $\mathbb{C}$ is said to *have pullbacks along* $\mathscr{M}$*-morphisms* if for each cospan $M \mathrel{-}m\mathrel{\rightarrow} B \mathrel{\leftarrow}f\mathrel{-} A$ with $m \in \mathscr{M}$, a pullback $M \mathrel{\leftarrow}g\mathrel{-} X \mathrel{-}n\mathrel{\rightarrow} A$ of $M \mathrel{-}m\mathrel{\rightarrow} B \mathrel{\leftarrow}f\mathrel{-} A$ exists in $\mathbb{C}$.

A.1 Functors, natural transformations, adjunctions

☼ Definition A.16 (Functor) Let $\mathbb{C}$ and $\mathbb{D}$ be categories. Then a *functor* $\mathcal{F}$ *from* $\mathbb{C}$ *to* $\mathbb{D}$, written $\mathcal{F}\colon \mathbb{C} \to \mathbb{D}$, is a pair $\mathcal{F} = \langle \mathcal{F}_0, \mathcal{F}_1 \rangle$ such that

- $\langle \mathcal{F}_0, \mathcal{F}_1 \rangle\colon \mathcal{U}\mathbb{C} \to \mathcal{U}\mathbb{D}$ is a graph morphism,
- for all objects $A \in \mathbb{C}$ the identity id_A is preserved, i.e. the equation

$$\mathcal{F}_1(\mathrm{id}_A^{\mathbb{C}}) = \mathrm{id}^{\mathbb{D}}_{\mathcal{F}_0(A)} \tag{7}$$

  holds, and
- for all objects $A, B, C \in \mathbb{C}$ and arrows $f\colon A \to B$ and $g\colon B \to C$ in $\mathbb{C}$ also the composition $f \fatsemi g$ is preserved:

$$\mathcal{F}_1(f \fatsemi_{\mathbb{C}} g) = \mathcal{F}_1(f) \fatsemi_{\mathbb{D}} \mathcal{F}_1(g). \tag{8}$$

☼

The subscripts on functors are usually suppressed. Functors can be used to congregate categories themselves into a (very) large category.

Example A.17 (Category of categories) The *category of categories* $\mathbb{Cat}$ has (small) categories as objects and functors between them as morphisms. Hence, functors are just category morphisms. The identity functor on a category $\mathbb{C}$ is $\mathrm{id}_{\mathbb{C}} \colon \mathbb{C} \to \mathbb{C}$ and acts as the identity on objects and morphisms, i.e. $\mathrm{id}_{\mathbb{C}}(A) = A$ for all $A \in \mathbb{C}$ and $\mathrm{id}_{\mathbb{C}}(f) = f$ for all $f \colon A \to B$ in $\mathbb{C}$. Further, composition of functors $\mathcal{F} \colon \mathbb{C} \to \mathbb{D}$ and $\mathcal{G} \colon \mathbb{D} \to \mathbb{E}$ is pointwise, i.e. $\mathcal{G} \circ \mathcal{F}(A) = \mathcal{G}(\mathcal{F}(A))$ for all $A \in \mathbb{C}$ and $\mathcal{G} \circ \mathcal{F}(f) = \mathcal{G}(\mathcal{F}(f))$ for all $f \colon A \to B$ in $\mathbb{C}$.

The canonical category with functors from a category $\mathbb{C}$ to a category $\mathbb{D}$ as objects is the functor category $[\mathbb{C}, \mathbb{D}]$. This category is the analogon of the set $B^A = \{f \mid f \colon A \to B \text{ in } \mathbb{Sets}\}$, which contains all functions from A to B – just replace $\mathbb{Sets}$ by $\mathbb{Cat}$. Its morphisms are natural transformations.

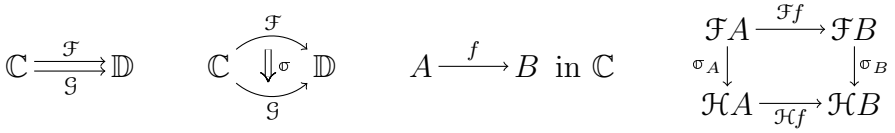


Figure 88: Natural transformation between two functors $\mathcal{F}, \mathcal{G} \colon \mathbb{C} \to \mathbb{D}$

Definition A.18 (Natural transformations) Let $\mathbb{C}$ and $\mathbb{D}$ be categories, and $\mathcal{F}, \mathcal{H} \colon \mathbb{C} \to \mathbb{D}$ be functors. A *natural transformation* $\sigma \colon \mathcal{F} \to \mathcal{H}$ is a family $\sigma = \{\sigma_A \colon \mathcal{F}(A) \to \mathcal{H}(A)\}_{A \in \mathbb{C}}$ such that the equation $\sigma_B \circ \mathcal{F}(f) = \mathcal{H}(f) \circ \sigma_A$ holds for all morphisms $f \colon A \to B$ in $\mathbb{C}$ (see Figure 88).

The *functor category* $[\mathbb{C}, \mathbb{D}]$ has functors $\mathcal{F} \colon \mathbb{C} \to \mathbb{D}$ and natural transformations between them as morphisms. The identity on $\mathcal{F}$ is

$$\mathrm{id}_{\mathcal{F}} = \{\mathrm{id}_{\mathcal{F}(A)} \colon \mathcal{F}(A) \to \mathcal{F}(A)\}_{A \in \mathbb{C}}$$

and the composition of natural transformation is pointwise, i.e. given $\sigma \colon \mathcal{F} \to \mathcal{H}$ and $\tau \colon \mathcal{H} \to \mathcal{K}$, their composition is

$$\tau \circ \sigma = \{\tau_A \circ \sigma_A \colon \mathcal{F}(A) \to \mathcal{K}(A)\}_{A \in \mathbb{C}}.$$

Example A.19 (Graphs as a functor category) Let $\cdot \rightrightarrows \cdot$ be the (up to isomorphism) unique category with two objects and two parallel arrows. Then $\mathbb{Graphs}$ is isomorphic to the functor category $[\cdot \rightrightarrows \cdot, \mathbb{Sets}]$, i.e. there

are functors $\mathcal{F}\colon \mathbb{G}\mathrm{raphs} \to [\cdot \rightrightarrows \cdot, \mathbb{S}\mathrm{ets}]$ and $\mathcal{G}\colon [\cdot \rightrightarrows \cdot, \mathbb{S}\mathrm{ets}] \to \mathbb{G}\mathrm{raphs}$ such that the equations $\mathcal{F} \circ \mathcal{G} = \mathrm{id}_{\mathbb{G}\mathrm{raphs}}$ and $\mathcal{G} \circ \mathcal{F} = \mathrm{id}_{[\cdot \rightrightarrows \cdot, \mathbb{S}\mathrm{ets}]}$ hold. More general, functor categories of the form $[\mathbb{C}, \mathbb{S}\mathrm{ets}]$ are essentially the same as (multi-sorted) algebras over signatures which only have unary function symbols.

Besides the composition of natural transformations between parallel functors, which is often called the *vertical composition*, there is also a horizontal composition.

Definition A.20 (Horizontal composition of functors) Let $\mathbb{C}, \mathbb{D}, \mathbb{E}$ be categories, let $\mathcal{F}, \mathcal{G}\colon \mathbb{C} \to \mathbb{D}$ and $\mathcal{H}, \mathcal{K}\colon \mathbb{D} \to \mathbb{E}$ be functors, and let $\sigma\colon \mathcal{F} \to \mathcal{G}$ and $\tau\colon \mathcal{H} \to \mathcal{K}$ be natural transformations.

Then the *horizontal composition of* σ *and* τ, written $\tau * \sigma\colon \mathcal{H} \circ \mathcal{F} \to \mathcal{K} \circ \mathcal{G}$, is defined as $\tau * \sigma := \{\tau_{\mathcal{G}(A)} \circ \mathcal{H}(\sigma_A)\}_{A \in \mathbb{C}}$.

Remark A.21 Alternatively one could also put $\tau * \sigma = \{\mathcal{K}(\sigma_A) \circ \tau_{\mathcal{F}(A)}\}_{A \in \mathbb{C}}$ since $\tau_{\mathcal{G}(A)} \circ \mathcal{H}(\sigma_A) = \mathcal{K}(\sigma_A) \circ \tau_{\mathcal{F}(A)}$ by naturality of σ and τ.

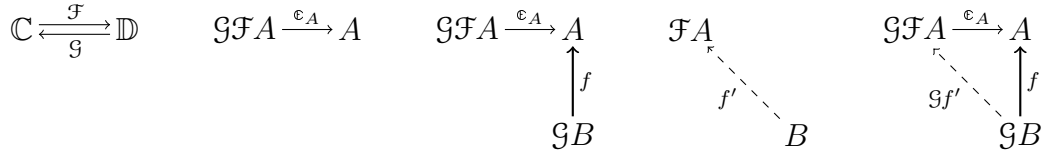


Figure 89: Adjunction $\mathcal{G} \dashv \mathcal{F}$ in terms of a co-unit $\mathfrak{e}\colon \mathcal{G} \circ \mathcal{F} \to \mathrm{id}$

Definition A.22 (Adjunction) Let $\mathbb{C}, \mathbb{D}$ be categories. Then an *adjunction* is a triple $\langle \mathcal{G}, \mathcal{F}, \mathfrak{e} \rangle$ where $\mathcal{F}\colon \mathbb{C} \to \mathbb{D}$ and $\mathcal{G}\colon \mathbb{D} \to \mathbb{C}$ are functors in opposite directions, and

$$\mathfrak{e} = \{\mathfrak{e}_A\colon \mathcal{G}(\mathcal{F}(A)) \to A\}_{A \in \mathbb{C}}\colon \mathcal{G} \circ \mathcal{F} \to \mathrm{id}_{\mathbb{C}},$$

which is called the *counit* of the adjunction, is a family of arrows in $\mathbb{C}$ such that each arrow $f\colon \mathcal{G}(B) \to A$ in $\mathbb{C}$ has a unique counterpart $f'\colon B \to \mathcal{F}(A)$ in $\mathbb{D}$ which satisfies $f = \mathfrak{e}_A \circ \mathcal{G}(f')$ where $A \in \mathbb{C}$ and $B \in \mathbb{D}$ are arbitrary objects (see Figure 90).

Given an adjunction $\langle \mathcal{G}, \mathcal{F}, \mathfrak{e} \rangle$, the functor $\mathcal{F}$ is called the *right adjoint (of* $\mathcal{G}$*)* and $\mathcal{G}$ is the *left adjoint (of* $\mathcal{F}$*)*. If $\mathcal{G}\colon \mathbb{D} \to \mathbb{C}$ is an embedding, i.e. if

$\mathcal{G}(B) = B$ for all $B \in \mathbb{D}$ and $\mathcal{G}(h) = h$ for all $h \in \mathrm{ar}(\mathbb{D})$, then the adjunction is called a *coreflection* and $\mathbb{D}$ is a coreflective subcategory of $\mathbb{C}$.

A fundamental fact of category theory is that if a right (or left) adjoint of a given functor exists, it is determined up to a canonical natural isomorphism. This means that, given a functor $\mathcal{G}\colon \mathbb{D} \to \mathbb{C}$ and two functors $\mathcal{F}, \mathcal{F}'\colon \mathbb{C} \to \mathbb{D}$ in the "opposite" direction such that both $\mathcal{G} \dashv \mathcal{F}$ and $\mathcal{G} \dashv \mathcal{F}'$ hold, there is a canonical natural isomorphism $\mathfrak{i}\colon \mathcal{F} \to \mathcal{F}'$ which witnesses that $\mathcal{F}$ and $\mathcal{F}'$ are isomorphic (in the functor category $[\mathbb{C}, \mathbb{D}]$).

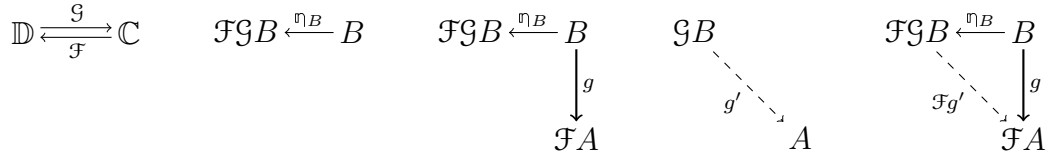


Figure 90: Adjunction $\mathcal{G} \dashv \mathcal{F}$ in terms of a unit $\eta\colon \mathrm{id} \to \mathcal{F} \circ \mathcal{G}$

Remark A.23 (Adjunctions via units and natural isomorphisms) An alternative characterization of an adjunction $\mathcal{G} \dashv \mathcal{F}$ works in the "opposite" direction and uses a natural transformation $\eta\colon \mathrm{id}_{\mathbb{D}} \to \mathcal{F} \circ \mathcal{G}$ (see Figure 90), which is called the unit of the adjunction. Moreover, adjunctions can be succinctly described as natural isomorphisms $\varphi\colon \mathbb{C}(\mathcal{G}(B), A) \cong \mathbb{D}(B, \mathcal{F}(A))$ between the homsets $\mathbb{C}(\mathcal{G}(B), A)$ and $\mathbb{D}(B, \mathcal{F}(A))$;

Example A.24 (Free graphs and diagrams as functors) Free categories of graphs actually arise by application of the left adjoint $\mathcal{L} \dashv \mathcal{U}$ of the underlying graph functor $\mathcal{U}\colon \mathbb{Cat} \to \mathbb{Graphs}$. In particular, each diagram $\mathcal{D}\colon J \to \mathbb{C}$ corresponds to a functor $\mathcal{D}'\colon \mathcal{L}(J) \to \mathbb{C}$.

Since also each functor $\mathcal{D}\colon \mathbb{J} \to \mathbb{C}$ is essentially the same as a diagram of shape $\mathcal{U}(\mathbb{J})$, diagrams are essentially the same as functors $\mathcal{D}\colon \mathbb{J} \to \mathbb{C}$. Moreover, cocones are essentially the same as natural transformations $\mathfrak{c}\colon \mathcal{D} \to \Delta_C^{\mathbb{J}}$ where for an object $C \in \mathbb{C}$, the functor $\Delta_C^{\mathbb{J}}\colon \mathbb{J} \to \mathbb{C}$ is constant at C, i.e. $\Delta_C^{\mathbb{J}}(i) = C$ for all $i \in \mathbb{J}$ and $\Delta_C^{\mathbb{J}}(e) = \mathrm{id}_C$ for all $e\colon i \to j$ in $\mathbb{J}$.

A.2 Slices and subobjects

Slice categories feature prominently in the unfolding of grammars, and also subobject posets occur passim.

Definition A.25 (Slice category) Let $\mathbb{C}$ be any category, and let $T \in \mathbb{C}$ be an object. Then the *slice category over* T, written $\mathbb{C}\downarrow T$, has all $\mathbb{C}$-morphisms

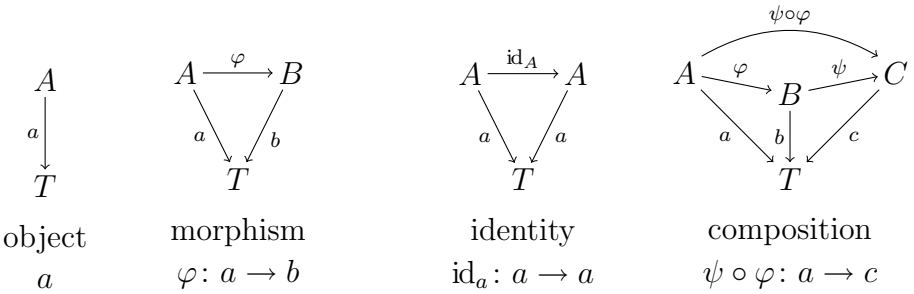


Figure 91: $\mathbb{C}\downarrow T$ – the slice category over T

with codomain T as objects, i.e. $\mathrm{ob}(\mathbb{C}\downarrow T) = \{a \in \mathrm{ar}(\mathbb{C}) \mid \mathrm{cd}\, a = T\}$; such an object is often written as $(A \text{–}a\text{→} \cdot) \in \mathbb{C}\downarrow T$ whenever T is clear from the context (but it is convenient to mention the domain A explicitly). Further, the morphisms between two $\mathbb{C}\downarrow T$-objects $A \text{–}a\text{→} \cdot$ and $B \text{–}b\text{→} \cdot$ are all $\mathbb{C}$-morphisms $\varphi\colon A \to B$ which satisfy the equation $a = b \circ \varphi$, i.e.

$$\mathbb{C}\downarrow T(a, b) = \{\varphi \in \mathbb{C}(\mathrm{dm}\, a, \mathrm{cd}\, b) \mid a = b \circ \varphi\}.$$

As illustrated in Figure 91, identities and composition are inherited from $\mathbb{C}$, in other words for every object $(a\colon A \to T) \in \mathbb{C}\downarrow T$, its identity is $\mathrm{id}_a^{\mathbb{C}\downarrow T} := \mathrm{id}_A^{\mathbb{C}}$, and for every pair of composable morphisms $\varphi, \psi \in \mathrm{ar}(\mathbb{C}\downarrow T)$, their composition is given by $\varphi \circ_{\mathbb{C}\downarrow T} \psi := \varphi \circ_{\mathbb{C}} \psi$. ☼

Similar to slice categories are subobject posets, which are generalizations of the power set lattice.

☼ Definition A.26 (Subobject poset) Let $T \in \mathbb{C}$ be an object. Then the *subobject poset* $\mathrm{Sub}(T) = \langle\, | \mathrm{Sub}(T)|, \sqsubseteq \rangle$ has isomorphism classes $[a\colon A \rightarrowtail T]$ of monomorphisms over T as elements. Formally, for any monomorphism $a\colon A \rightarrowtail T$ define

$$[a] := \left\{ b\colon B \rightarrowtail T \;\middle|\; \begin{array}{l} \exists\, \mathbb{C}\text{-isomorphism } i\colon A \text{–}\cong\text{→} B. \\ \quad a = b \circ i \end{array} \right\} \qquad \begin{array}{ccc} A & \dashrightarrow^{\cong} & B \\ & {\scriptstyle a}\searrow \quad \swarrow{\scriptstyle b} & \\ & T & \end{array},$$

which is the *subobject represented by* a. Further, given two monomorphisms $a\colon A \rightarrowtail T$ and $b\colon B \rightarrowtail T$, the inclusion $[a] \sqsubseteq [b]$ holds if there exists some morphism $j\colon A \to B$ in $\mathbb{C}$ such that $a = b \circ j$. Summarizing, the subobject poset over T is $\mathrm{Sub}(T) = \langle\, \{[a\colon A \rightarrowtail T] \mid a \text{ monic }\}, \sqsubseteq \rangle$. ☼

A basic operation from a slice categories $\mathbb{C}\downarrow T$ to another one $\mathbb{C}\downarrow X$ is (post-)composition with a morphism $f\colon T \to X$.

☼ Definition A.27 (Postcomposition functor) Let $f\colon T \to X$ be a morphism in a category $\mathbb{C}$. Then the morphism f induces a *post-composition with f* functor $\Sigma_f\colon \mathbb{C}\downarrow T \to \mathbb{C}\downarrow X$ which is defined by $\Sigma_f(a) = f \circ a$ for all $a \in \mathbb{C}\downarrow T$ and $\Sigma_f(\varphi) = \varphi$ for all $\varphi\colon a \to b$ in $\mathbb{C}\downarrow T$. ☼

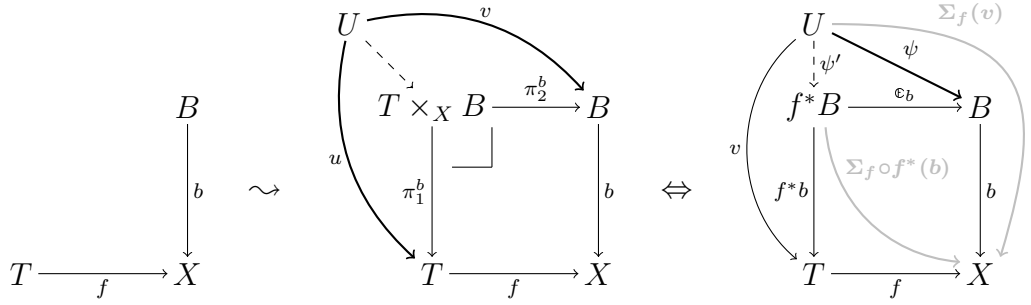


Figure 92: Illustration of Lemma A.28

◌ Lemma A.28 (Pullbacks and right adjoints to post-compositon) A category $\mathbb{C}$ has pullbacks if and only if for each morphism $f\colon T \to X$ the functor Σ_f has a right adjoint $\Sigma_f \dashv f^*$.

Proof. The proof is a straightforward calculation using the characterization of adjunctions in terms of counits (see also Figure 92). □

Hence, right adjoints of Σ_f are usually known as pullback functors; they induce preimage functors between the respective subobject posets.

☼ Definition A.29 (Preimages) Assuming choice, there is an "inclusion"-functor $\mathcal{J}\colon \mathrm{Sub}(T) \to \mathbb{C}\downarrow T$ which picks a representative $\mathcal{J}t$ of each subobject $t \in \mathrm{Sub}(T)$ such that $t = [\mathcal{J}t]$. Then, for a given morphism $f\colon T \to X$ the *preimage functor* $f^{-1}\colon \mathrm{Sub}(X) \to \mathrm{Sub}(T)$ maps each subobject $t \in \mathrm{Sub}(T)$ to $[f^*(\mathcal{J}t)]$ where $f^*\colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow T$ is right adjoint to Σ_f, i.e. $\Sigma_f \dashv f^*$. ☼

A.3 Miscellaneous

☼ Definition A.30 (Admissible sets of monomorphisms) Let $\mathbb{C}$ be a category, and let $\mathscr{M} \subseteq \mathrm{ar}(\mathbb{C})$ be a class of $\mathbb{C}$-monomorphisms, i.e. each element $m \in \mathscr{M}$ is monic in $\mathbb{C}$; each $m \in \mathscr{M}$ is called an *$\mathscr{M}$-monomorphism.*

If $\mathbb{C}$ has pullbacks along $\mathscr{M}$-monomorphisms, a class of monomorphisms $\mathscr{M}$ is *stable (under pullback)* if for each $\mathbb{C}$-cospan $B \xrightarrow{f} A \xleftarrow{m} C$ with $m \in \mathscr{M}$

and each pullback $B \leftarrow m' \prec D - f' \rightarrow C$ of $B - f \rightarrow A \leftarrow m \prec C$, also the monomorphism m' is in $\mathscr{M}$ (see also Figure 93).

Finally $\mathscr{M}$ is an *admissible* class of monomorphisms, if each $\mathbb{C}$-identity $\mathrm{id}_A \colon A \to A$ is contained in $\mathscr{M}$, i.e. $\mathrm{id}_A \in \mathscr{M}$, and further $\mathscr{M}$ is closed under composition, i.e. whenever $\langle m, k \rangle \in \mathscr{M} \times \mathscr{M}$ is a composable pair in $\mathbb{C}$, then also $m \circ k \in \mathscr{M}$. ☼

$$m \in \mathscr{M} \quad \& \quad \begin{array}{ccc} D & \xrightarrow{f'} & C \\ {\scriptstyle m'}\downarrow & \lrcorner & \downarrow{\scriptstyle m} \\ B & \xrightarrow[f]{} & A \end{array} \quad \Rightarrow \quad m' \in \mathscr{M}$$

Figure 93: Stable class of monomorphisms

☼ Definition A.31 ($\mathscr{M}$-categories and categories of partial maps) Let $\mathbb{C}$ be a category with pullbacks and let $\mathscr{M}$ be an admissible class of monomorphisms in $\mathbb{C}$. Then $\langle \mathbb{C}, \mathscr{M} \rangle$ is an $\mathscr{M}$-category.

Given an $\mathscr{M}$-category $\langle \mathbb{C}, \mathscr{M} \rangle$, the *category* $\mathrm{Par}(\mathbb{C}, \mathscr{M})$ *of* $\mathscr{M}$*-partial maps* has objects $\mathrm{ob}\big(\mathrm{Par}(\mathbb{C}, \mathscr{M})\big) = \mathrm{ob}(\mathbb{C})$ and homsets

$$\mathrm{Par}(\mathbb{C}, \mathscr{M})(A, B) = \left\{ [m_{(X)} f\rangle \;\middle|\; \begin{array}{l} m \in \mathscr{M} \;\& \\ m \colon X \to A \;\& \\ f \colon X \to B \text{ in } \mathbb{C} \end{array} \right\} \qquad A \xleftarrow{m} X \xrightarrow{f} B$$

where

$$[m_{(X)} f\rangle = \left\{ \langle A \leftarrow n \prec Y, Y \xrightarrow{g} B\rangle \;\middle|\; \begin{array}{l} \exists\ \mathbb{C}\text{-isomorphism } i \colon X \xrightarrow{\cong} Y. \\ \quad m = n \circ i \;\&\; i \fatsemi g = f \end{array} \right\}.$$

Further the identity of an object $A \in \mathrm{Par}(\mathbb{C}, \mathscr{M})$ is $\mathrm{id}_A = [\mathrm{id}^{\mathbb{C}}_{A\,(A)} \mathrm{id}^{\mathbb{C}}_A\rangle$, and the composition of two partial maps $[m_{(X)} f\rangle \colon A \to B$ and $[k_{(Z)} h\rangle \colon B \to C$ is based on some arbitrary pullback $X \leftarrow p \prec U - q \rightarrow Z$ of $X - f \rightarrow B \leftarrow k \prec C$ and then defined as $[m_{(X)} f\rangle \fatsemi [k_{(Z)} h\rangle = [m \circ p_{(U)} q \fatsemi h\rangle$ (see also Figure 94). ☼

The following definition is based on [Adámek et al., 1990, Definition 14.1].

☼ Definition A.32 (Factorization system) Let $\mathscr{E}$ and $\mathscr{M}$ be two classes of morphisms in a category $\mathbb{C}$. Then $(\mathscr{E}, \mathscr{M})$ is called a *factorization system* for all morphisms in $\mathbb{C}$ provided that

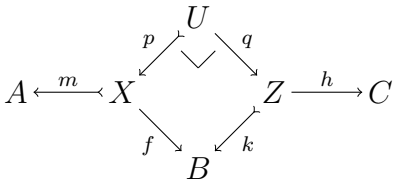


Figure 94: Illustration of composition of partial maps

1. both $\mathscr{E}$ and $\mathscr{M}$ are closed under composition with isomorphisms;
2. the category $\mathbb{C}$ has $(\mathscr{E},\mathscr{M})$-factorizations (of morphisms), i.e. each morphism f in $\mathbb{C}$ has a factorization $f = m \circ e$ with $e \in \mathscr{E}$ and $m \in \mathscr{M}$;

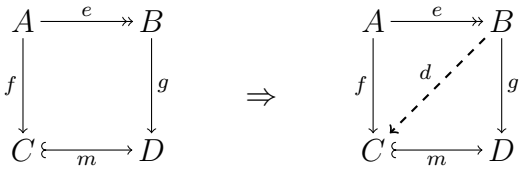


Figure 95: Unique diagonalization property

3. the category $\mathbb{C}$ has the unique $(\mathscr{E},\mathscr{M})$-diagonalization property, i.e. for each commutative square of the form shown on the left in Figure 95, with $e \in \mathscr{E}$ and $m \in \mathscr{M}$ there exists a unique diagonal, i.e. there is a unique morphism d such that the diagram on the right in Figure 95 commutes ($d \circ e = f$ and $m \circ d = g$).

☼

☼ Definition A.33 (Comma categories) Let $\mathbb{E}$,$\mathbb{C}$, and $\mathbb{D}$ be categories and let $\mathcal{F}\colon \mathbb{E} \to \mathbb{C}$ and $\mathcal{G}\colon \mathbb{D} \to \mathbb{C}$ be functors. Then the *comma category* $\mathcal{F}{\downarrow}\mathcal{G}$ has as objects all triples $\langle E, D, f\rangle \in \mathrm{ob}(\mathbb{E}) \times \mathrm{ob}(\mathbb{D}) \times \mathrm{ar}(\mathbb{C})$ such that $f\colon \mathcal{F}(E) \to \mathcal{G}(D)$, and the homset between two objects $\langle E, D, f\rangle$ and $\langle E', D', f'\rangle$ is the collection of all pairs of morphisms $\langle k\colon E \to E', h\colon D \to D'\rangle \in \mathrm{ar}(\mathbb{E}) \times \mathrm{ar}(\mathbb{D})$ satisfying the equation $\mathcal{F}(k) \fatsemi f' = f \fatsemi \mathcal{G}(h)$ (see also Figure 96). ☼

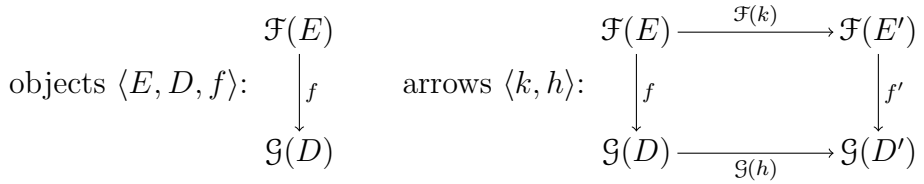


Figure 96: The comma category $\mathcal{F}\downarrow\mathcal{G}$ for a cospan of functors $\mathbb{E} -\mathcal{F}\rightarrow \mathbb{C} \leftarrow\mathcal{G}- \mathbb{D}$

Results about partial Van Kampen colimits

B

This section provides the general results on on partial Van Kampen colimits. The main new concept is related to the generalization of the assumption in the characterization of (partial) Van Kampen squares that the back faces of a certain cube are pullbacks. This is achieved by use of the concept of a cartesian (diagram) morphism.

☼ Definition B.1 (Cartesian diagram morphism) Let $\mathcal{D}, \mathcal{E} \colon I \to \mathbb{C}$ be diagrams, and let $\mathbb{f} \colon \mathcal{E} \to \mathcal{D}$ be a diagram morphism. Then $\mathbb{f}$ is *cartesian* if $\cdot \leftarrow_{\mathcal{E}_E(e)} - \cdot - \rightarrow_{\mathbb{f}_i} \cdot$ is a pullback of $\cdot - \rightarrow_{\mathbb{f}_j} \cdot \leftarrow_{\mathcal{D}_E(e)} - \cdot$ for each $e \colon j \leftarrow i$ in I (see Figure 97). ☼

Figure 97: Cartesian diagram morphism

With this concept at hand, pullback merging and $\mathscr{M}$-universality of PVK-colimits can be derived as follows.

◌ Lemma B.2 (PVK assumming $\mathscr{M}ono$-universality) Let I be a graph, let $\mathbb{C}$ be a category with pullbacks along (mono-)morphisms, and let $\mathbb{C}$ have colimits of shape I that are pullback stable (along monomorphisms). Finally let $\mathcal{D} \colon I \to \mathbb{C}$ be a diagram and let $\mathbb{d} \colon \mathcal{D} \to \Delta_D^I$ be its colimit.

Then the colimit $\mathbb{d}$ is preserved by the graphing functor $\Gamma_{\mathscr{M}} \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$ if and only if for each diagram $\mathcal{E} \colon I \to \mathbb{C}$ with colimit $\mathbb{e} \colon \mathcal{E} \to \Delta_E^I$ and each monic cartesian diagram morphism $\mathbb{m} \colon \mathcal{E} \rightarrowtail \mathcal{D}$, the unique mediating morphism $u \colon \mathbb{e} \to \mathbb{d} \circ \mathbb{m}$ is monic and $\cdot \leftarrow_{\mathbb{e}_j} - \cdot \succ_{\mathbb{m}_j} \cdot$ is a pullback of $\cdot \succ_u \cdot \leftarrow_{\mathbb{d}_j} - \cdot$ for each $j \in V_I$ (see also Figure 98).

Proof. First, assume that the graphing functor $\Gamma^{\mathbb{C}} \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$ preserves the colimit $\mathbb{d}$. Further let $\mathcal{E} \colon I \to \mathbb{C}$ be a diagram, let $\mathbb{e} \colon \mathcal{E} \to \Delta_E^I$ be its colimit, and let $\mathbb{m} \colon \mathcal{E} \rightarrowtail \mathcal{D}$ be a cartesian diagram morphism. Finally let $u \colon E \to D$ the unique $\mathbb{C}$-morphism satisfying $u \circ \mathbb{e}_i = \mathbb{d}_i \circ \mathbb{m}_i$ for each $i \in V_I$. This implies that the family $\{[\mathbb{m}_i, \mathbb{e}_i\rangle \colon \mathcal{D}_V(i) \rightharpoonup E\}_{i \in V_I}$ is a $\mathrm{Par}(\mathbb{C})$-cocone of the diagram $\Gamma^{\mathbb{C}} \circ \mathcal{D}$ (see the left hand diagram in Figure 98). As the graphing functor $\Gamma^{\mathbb{C}}$ preserves the colimit $\mathbb{d}$, there exists a partial map $[w, r\rangle \colon D \rightharpoonup E$ such that

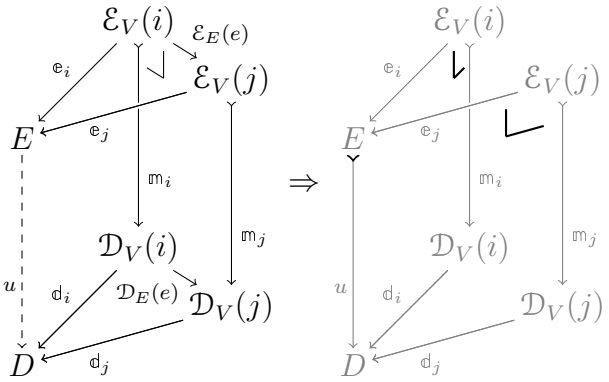
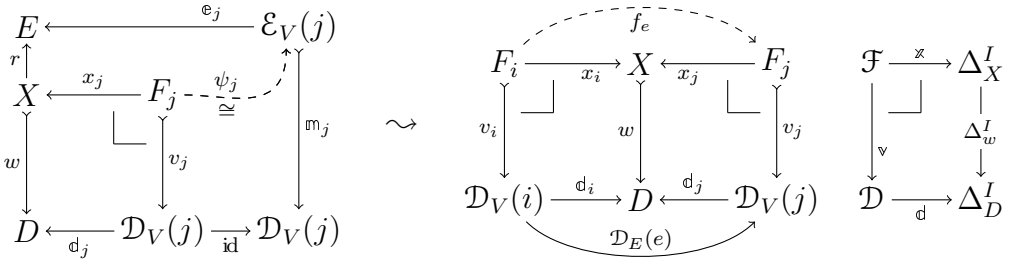


Figure 98: Illustration of Lemma B.2

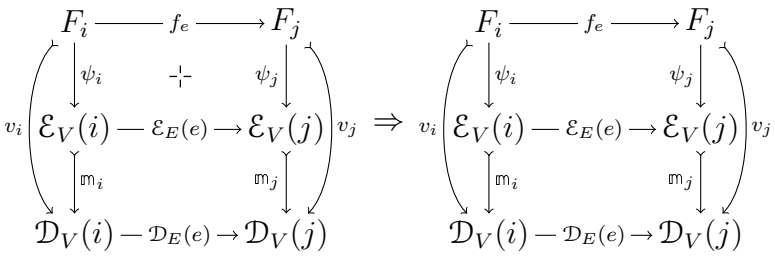
$[w,r\rangle \circ \mathbb{d}_j = [\mathbb{m}_j,\mathbb{e}_j\rangle$ for all $j \in V_I$, as shown in the left one of the following diagrams.



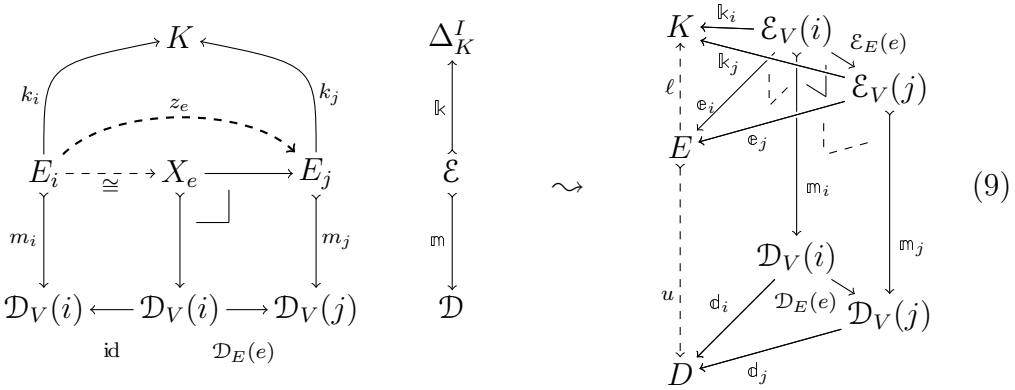
Moreover, as $\mathcal{D}$ is stable under pullback (along monomorphisms), the family $\mathbb{x} := \{x_i \colon F_i \to X\}_{i \in V_I}$ is a colimit of the diagram $\mathcal{F} \colon I \to \mathbb{C}$ given by $\mathcal{F}_V(i) = F_i$ and $\mathcal{F}_E(e) = f_e$, as shown on the right in the last display. Now it remains to show that the family $\mathbb{y} := \{\psi_j\}_{j \in V_I}$ is a diagram (iso-)morphism $\mathbb{y} \colon \mathcal{F} \xrightarrow{\cong} \mathcal{E}$ because then $\mathbb{e} \circ \mathbb{y}$ and $\mathbb{x}$ are colimits of the same diagram, which in turn implies that $r \colon X \to E$ is an isomorphism.

That $\mathbb{y} \colon \mathcal{F} \xrightarrow{\cong} \mathcal{E}$ is a diagram morphism is easily verified as follows

because $\mathbb{m}_j$ is monic.[82]



For the converse,. let $\mathbb{h} = \{[m_{j(E_j)}k_j\rangle\colon \mathcal{D}_V(j) \rightarrowtail K\}_{j\in V_I}$ be a $\mathrm{Par}(\mathbb{C})$-cocone of $\Gamma^{\mathbb{C}} \circ \mathcal{D}$. This means that $[m_i,k_i\rangle = [m_j,k_j\rangle \circ \mathcal{D}_E(e)$ holds for each $i \xrightarrow{e} j$ in I. Now define a diagram $\mathcal{E}\colon I \to \mathbb{C}$ by setting $\mathcal{E}_V(i) := E_i$ for each $i \in I$ and putting $\mathcal{E}_E(e) := z_e\colon E_i \to E_j$ for each $e\colon i \to j$ in I where the $z_e\colon E_i \to E_j$ are constructed as shown in the left one of the following diagrams.



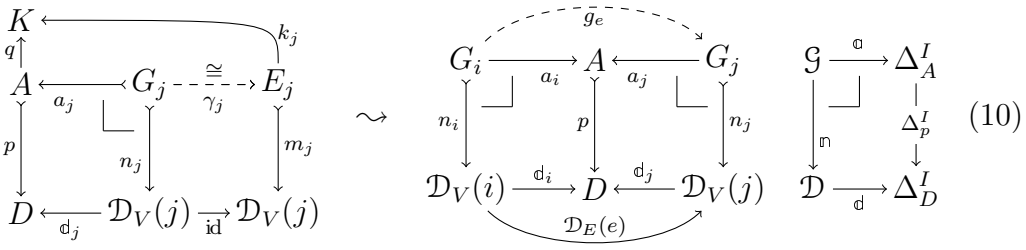
The families $\mathbb{m} := \{m_i\colon \mathcal{E}_V(i) \rightarrowtail \mathcal{D}_V(i)\}_{i\in V_I}$ and $\mathbb{k} := \{k_i\colon \mathcal{E}_V(i) \to K\}_{i\in V_I}$ form a cartesian diagram morphism $\mathbb{m}\colon \mathcal{E} \rightarrowtail \mathcal{D}$ and cocone $\mathbb{k}\colon \mathcal{E} \to \Delta^I_K$, respectively. Now, as illustrated in the preceeding display on the right, construct the $\mathbb{C}$-colimit $\mathbb{e}\colon \mathcal{E} \to \Delta^I_E$, which results in mediating morphisms $\ell\colon \mathbb{e} \to \mathbb{k}$ and $u\colon \mathbb{e} \to \mathbb{d} \circ \mathbb{m}$. However, using the assumption,[83] u is a monomorphism and $\cdot \leftarrow{\mathbb{e}_j}- \cdot \rightarrowtail{\mathbb{m}_j}\to \cdot$ a pullback of $\cdot \rightarrowtail{u}\to \cdot \leftarrow{\mathbb{d}_j}- \cdot$ for each $j \in V_I$; in other words $[u,\ell\rangle\colon \mathbb{d} \rightarrowtail \mathbb{h}$ is a mediating morphism.[84]

[82] Recall that the puncture sign '-|-' indicates that *only* the smallest surrounding region is not assumed to commute.

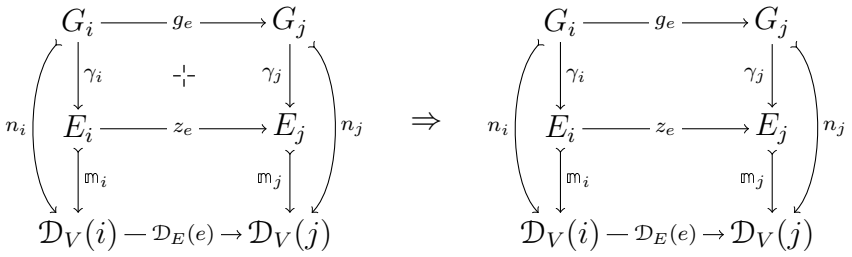
[83] The (implicit) assumption is that for each diagram $\mathcal{E}\colon I \to \mathbb{C}$, colimit $\mathbb{e}\colon \mathcal{E} \to \Delta^I_E$, and monic cartesian morphism $\mathbb{m}\colon \mathcal{E} \rightarrowtail \mathcal{D}$, the unique mediating morphism $u\colon \mathbb{e} \to \mathbb{d} \circ \mathbb{m}$ is monic and $\cdot \leftarrow{\mathbb{e}_j}- \cdot \rightarrowtail{\mathbb{m}_j}\to \cdot$ is a pullback of $\cdot \rightarrowtail{u}\to \cdot \leftarrow{\mathbb{d}_j}- \cdot$ for each $j \in V_I$.

[84] A reader with a mind for such things will easily verify that this argument is independent of the choice of the representants of the $\mathbb{h}_i\colon \mathcal{D}_V(i) \rightarrowtail K$.

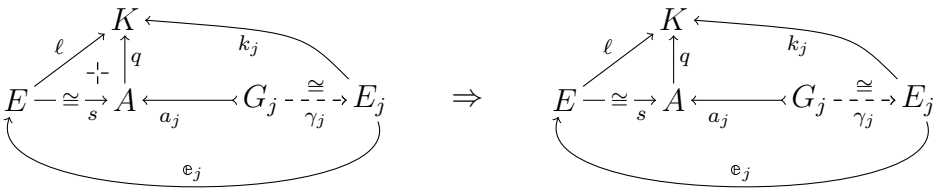
It remains to show that $[u,\ell\rangle\colon \mathbb{d} \rightharpoonup \mathbb{h}$ is the *unique* mediating morphism. Hence let $[p_{(A)}q\rangle\colon \mathbb{d} \rightharpoonup \mathbb{h}$ be another mediating morphism, which means that $[p_{(A)}q\rangle\colon D \rightharpoonup K$ satisfies $[p,q\rangle \circ \mathbb{d}_j = [m_j,k_j\rangle$ for all $j \in V_I$.



Now, proceeding as in the first part of the proof, pull back $\mathcal{D}$ along (the monomorphism) p obtaining a diagram $\mathcal{G}\colon I \to \mathbb{C}$ with $\mathcal{G}_V(l) = G_l$ and $\mathcal{G}_E(e) = g_e\colon G_i \to G_j$ (for each $l \in V_I$ and $e\colon i \to j$ in I), a cartesian monic diagram morphism $\mathbb{n} := \{n_i\colon G_i \to \mathcal{D}_V(i)\}_{i\in V_I}\colon \mathcal{G} \rightarrowtail \mathcal{D}$, and also a colimit $\mathbb{a} := \{a_i\colon G_i \to A\}_{i\in V_I}\colon \mathcal{G} \to \Delta_A^I$, as shown in the preceeding display. The family $\mathbb{g} := \{\gamma_i\colon G_i \to E_i\}_{i\in V_I}$ of isomorphisms is actually a diagram morphism $\mathbb{g}\colon \mathcal{G} \to \mathcal{E}$.



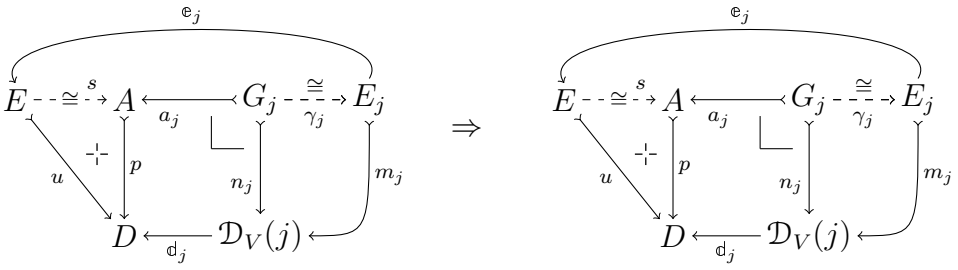
The candidate for an ismorphism $s\colon E \xrightarrow{\cong} A$ witnessing $[p,q\rangle = [u,\ell\rangle$, is the mediating isomorphism $s\colon \mathbb{e}\circ\mathbb{g} \to \mathbb{a}$, which exists because $\mathbb{e}\circ\mathbb{g}$ is a colimit of $\mathcal{G}$ as $\mathbb{g}$ is an isomorphism; i.e. the equation $s \circ \mathbb{e}_j \circ \mathbb{g}_j = \mathbb{a}_j$ holds for all $j \in V_I$. Further the equalities $q \circ a_j = k_j \circ \gamma_j$ and $k_j = \ell \circ e_j$ for all $j \in V_I$ (taken from (16) and (17), respectively), yield the following situation.



Now the equation $q \circ s = \ell$ follows from the fact that the family $\{\mathbb{e}_j \circ \gamma_j\}_{j \in V_I}$ is jointly epic.

$$\begin{aligned} q \circ \underline{s \circ \mathbb{e}_j \circ \gamma_j} &= \underline{q \circ a_j} \\ &= \underline{k_j} \circ \gamma_j \\ &= \ell \circ \mathbb{e}_j \circ \gamma_j \qquad \{\mathbb{e}_j \circ \gamma_j\}_{j \in V_I} \text{ jointly epic} \Rightarrow q \circ s = \ell \end{aligned}$$

Hence it remains to show that $p \circ s = u$, which is done in a similar way using again that the family $\{\mathbb{e}_j \circ \gamma_j\}_{j \in V_I}$ is jointly epic.



For all $j \in V_I$, the equations $s \circ \mathbb{e}_j \circ \gamma_j = a_j$ hold by construction of s as mediating morphism $s \colon \mathbb{e} \circ \mathbb{g} \to \mathbb{a}$, the equalities $u \circ j = j \circ m_j$ can be read off from the right diagram in (17); commatativity of the two "inner" regions in the preceeding display, i.e. $p \circ a_j = \mathbb{d}_j \circ n_j$ and $n_j = \gamma_j \circ m_j$, follows from commutativity of the left hand diagram in (16). Now diagram chaising yields the following equations.[85]

$$\begin{aligned} p \circ \underline{s \circ \mathbb{e}_j \circ \gamma_j} &= p \circ a_j \\ &= \mathbb{d}_j \circ \underline{n_j} \\ &= \underline{\mathbb{d}_j \circ m_j} \circ \gamma_j \\ &= u \circ \mathbb{e}_j \circ \gamma_j \qquad \{\mathbb{e}_j \circ \gamma_j\}_{j \in V_I} \text{ jointly epic} \Rightarrow p \circ s = u \end{aligned}$$

□

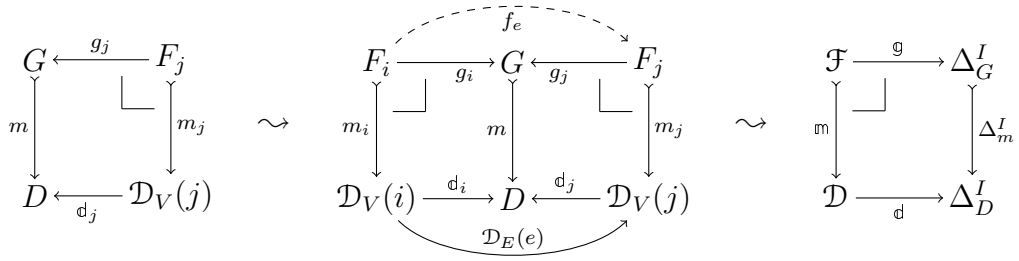
Lemma B.3 (*Mono*-universality of hereditary colimits) Let I be a graph, let $\mathbb{C}$ be a category with pullbacks along monomorphisms, let $\mathrm{Par}(\mathbb{C})$ be the associated partial map category, and let $\mathbb{C}$ have colimits of shape I.

Then colimits of shape I are *Mono*-universal if they are preserved by the graphing functor $\Gamma_{\mathscr{M}} \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$.

[85] Again, a reader with a mind for such things will readily verify that this proof is independent from the choices of representants of partial maps.

Proof. The premise is that colimits of shape I are preserved by the graphing functor $\Gamma^{\mathbb{C}}\colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$. Now let $\mathcal{D}\colon I \to \mathbb{C}$ be a diagram and further let $\mathbb{d}\colon \mathcal{D} \to \Delta^I_D$ be its colimit; the goal is to show that the diagram $\mathbb{d}$ is mono-universal.

To show the latter, let $m\colon G \rightarrowtail D$ be a $\mathbb{C}$-monomorphism, and construct a diagram $\mathcal{F}\colon I \to \mathbb{C}$ with cocone $\mathbb{g}\colon \mathcal{F} \to \Delta^I_G$ by pulling back the diagram $\mathcal{D}$ along the monomorphism m. Now it remains to show that the resulting cocone $\mathbb{g}\colon \mathcal{F} \to \Delta^I_G$ is a $\mathbb{C}$-colimit of the diagram $\mathcal{F}$. The construction of $\mathcal{F}$ and $\mathbb{g}$ is summarized in the following display.



That $\mathbb{g}\colon \mathcal{F} \to \Delta^I_G$ is a colimit of $\mathcal{F}$ can be proved using the fact that the partial map $[m,\mathrm{id}_G\rangle\colon D \rightharpoonup G$ is the *unique* mediating arrow from the $\mathrm{Par}(\mathbb{C})$-colimit $\mathbb{d}\colon \Gamma^{\mathbb{C}} \circ \mathcal{D} \to \Delta^I_D$ to the cocone[86] $\mathbb{p} := \{[m_i,\mathbb{g}_i\rangle\colon \mathcal{D}_V(i) \rightharpoonup G\}_{i\in V_I}$ in $\mathrm{Par}(\mathbb{C})$. The partial map $[m,\mathrm{id}_G\rangle\colon D \rightharpoonup G$ is the unique mediating arrow as, by assumption, the graphing functor $\Gamma^{\mathbb{C}}$ preserves the colimit $\mathbb{d}$; the partial map $[m,\mathrm{id}_G\rangle\colon D \rightharpoonup G$ in $\mathrm{Par}(\mathbb{C})$ can be described in one line as follows.

$$[m,\mathrm{id}_G\rangle = (\imath)p\colon D \rightharpoonup G.\ \big[\forall i \in V_I.\ p \circ \mathbb{d}_i = [m_i,\mathbb{g}_i\rangle\big] \tag{11}$$

where $(\imath)x.\,[\Phi]$ is the Russellian description operator which denotes the unique x satisfying Φ (if such an object exists).

Let $\mathbb{h}\colon \mathcal{F} \to \Delta^I_H$ be the $\mathbb{C}$-colimit of $\mathcal{F}$. This yields a mediating $\mathbb{C}$-morphism $g\colon \mathbb{h} \to \mathbb{g}$.

$$g\colon H \to G \qquad \forall i \in V_I.\quad \mathbb{g}_i = g \circ \mathbb{h}_i \tag{12}$$

As $\mathbb{d}\colon \Gamma^{\mathbb{C}} \circ \mathcal{D} \to \Delta^I_G$ is a $\mathrm{Par}(\mathbb{C})$-colimit, there exists a unique partial map $[n,k\rangle\colon D \rightharpoonup H$ such that $[n,k\rangle \circ \mathbb{d}_i = [m_i,\mathbb{h}_i\rangle$ holds for all $i \in V_I$.

$$[n,k\rangle = (\imath)q\colon D \rightharpoonup H.\ \big[q \circ \mathbb{d}_i = [m_i,\mathbb{h}_i\rangle\big] \tag{13}$$

The constructions made so far are summarized in the left hand diagram in Figure 99. Below it will be shown that there is an isomorphism $N \mathrel{-\!\!\cong\!\!\to} G$

[86]The reader might want to check that $\mathbb{p} := \{[m_i,\mathbb{g}_i\rangle\colon \mathcal{D}_V(i) \rightharpoonup G\}_{i\in V_I}$ is actually a $\mathrm{Par}(\mathbb{C})$-cocone of $\Gamma^{\mathbb{C}} \circ \mathcal{D}\colon I \to \mathrm{Par}(\mathbb{C})$.

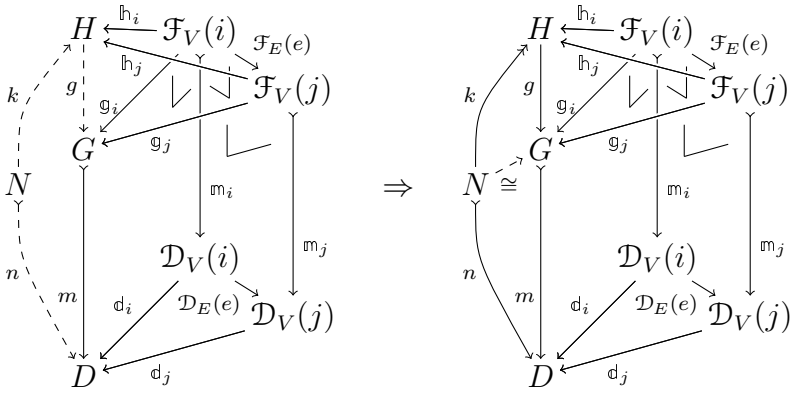


Figure 99: Illustration of the proof idea of Lemma B.3

witnessing that $[n,k \fatsemi g\rangle = [m,\mathrm{id}_G\rangle$, which implies that k is split monic; further the morphism k will turn out to be epic as well since the family $\{\mathbb{h}_i\}_{i\in V_I}$ is jointly epic (see the right hand diagram in Figure 99).

Returning to the actual proof, Equation (13) implies $[n,k\rangle \circ \mathbb{d}_j = [m_j,\mathbb{h}_j\rangle$ for all $j \in V_I$. As a consequence,

$$\begin{aligned}
\underline{[n,(k \fatsemi g)\rangle} \circ \mathbb{d}_j &= \\
g \circ [n,k\rangle \circ \mathbb{d}_j &= g \circ \underline{[m_j,\mathbb{h}_j\rangle} \\
&= \underline{g \circ \mathbb{h}_j} \circ [m_j,\mathrm{id}_{\mathcal{F}(j)}\rangle \\
&= \underline{g \circ \mathbb{h}_j} \circ [m_j,\mathrm{id}_{\mathcal{F}(j)}\rangle && \text{(by Equation (12))} \\
&= \mathbb{g}_j \circ [m_j,\mathrm{id}_{\mathcal{F}(j)}\rangle \\
&= [m_j,\mathbb{g}_j\rangle \\
&= [m,\mathrm{id}_G\rangle \circ \mathbb{d}_j && \text{(by construction)}
\end{aligned}$$

for each $j \in V_I$. Hence the equality $[n,(k \fatsemi g)\rangle = [m,\mathrm{id}_G\rangle$ follows from Equation (11), i.e. there is an isomorphism $\varphi\colon N \to G$ satisfying

$$k \fatsemi g = \varphi \fatsemi \mathrm{id}_G \qquad \text{and} \qquad n = \varphi \fatsemi m. \tag{14}$$

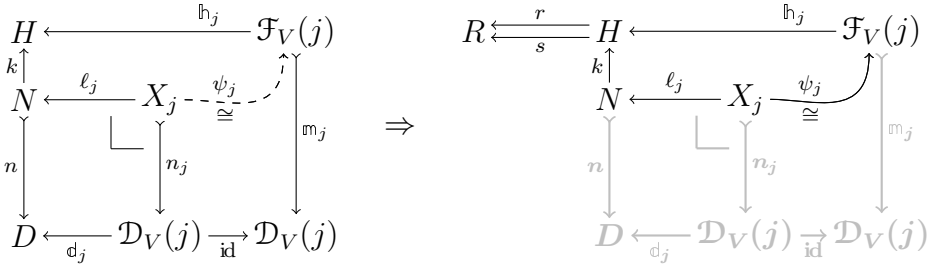
The first of the latter equations implies that k is split monic.

The last substantial step of the proof consists in showing that k is also epic, as then the desired follows easily as detailed below. To demonstrate that k is epic let $r, s\colon H \to R$ be $\mathbb{C}$-morphisms satifying $r \circ k = s \circ k$. Next construct for each $j \in V_I$ a pullback $N \xleftarrow{\ell_j} X_j \xrightarrow{n_j} \mathcal{D}_V(j)$ of $N \xrightarrow{n} D \xleftarrow{\mathbb{d}_j} \mathcal{D}_V(j)$;

from Equation (13) it follows that there are isomorphisms $\psi_j \colon X_j \xrightarrow{\cong} \mathcal{F}_V(j)$ satisfying

$$k \circ \ell_j = \mathbb{h}_j \circ \psi_j \qquad \text{and} \qquad \psi_j \fatsemi \mathbb{m}_j = n_j \tag{15}$$

(see the left hand diagram in the display below).



As the family $\{\mathbb{h}_j\}_{j \in V_I}$ is jointly epic in $\mathbb{C}$, also the family $\{\mathbb{h}_j \circ \psi_j\}_{j \in V_I}$ is so. Hence derive the chain of equations

$$\begin{aligned} r \circ \mathbb{h}_j \circ \psi_j &= r \circ k \circ \ell_j && \text{(see the previous display)} \\ &= s \circ k \circ \ell_j && \text{(by assumption about } s \text{ and } r\text{)} \\ &= s \circ \mathbb{h}_j \circ \psi_j && \text{(see the previous display)} \end{aligned}$$

to prove that r is equal to s, whence k is not only split monic but also epic, and therefore k is an isomorphism. This in turn implies that also g is an ismorphism and hence the cocone $\mathbb{g} \colon \mathcal{F} \to \Delta_G^I$ is isomorphic to the colimit $\mathbb{h} \colon \mathcal{F} \to \Delta_H^I$, which means that $\mathbb{g} \colon \mathcal{F} \to \Delta_G^I$ is a colimit of $\mathcal{F}$.

□

Theorem B.4 (Partial Van Kampen characterization) Let I be a graph, let $\mathbb{C}$ be a category with pullbacks along monomorphisms, let $\mathrm{Par}(\mathbb{C})$ be the associated partial map category, and let $\mathbb{C}$ have colimits of shape I. Finally let $\mathcal{D} \colon I \to \mathbb{C}$ be a diagram and let $\mathbb{d} \colon \mathcal{D} \to \Delta_D^I$ be its colimit.

Then the colimit $\mathbb{d}$ is partial Van Kampen, i.e. $\mathbb{d}$ is preserved by the graphing functor $\Gamma_{\mathscr{M}} \colon \mathbb{C} \to \mathrm{Par}(\mathbb{C})$, if and only if for each diagram $\mathcal{E} \colon I \to \mathbb{C}$, monic cartesian diagram morphism $\mathbb{m} \colon \mathcal{E} \rightarrowtail \mathcal{D}$, cocone $\mathbb{e} \colon \mathcal{E} \to \Delta_E^I$ and mediating morphism $u \colon \mathbb{e} \to \mathbb{d} \circ \mathbb{m}$ the following are equivalent:

(a) The cocone $\mathbb{e}$ is a colimit;

(b) The span $E \xleftarrow{\mathbb{e}_i} \mathcal{E}_V(i) \xrightarrow{\mathbb{m}_i} \mathcal{D}_V(i)$ is a pullback of $E \xrightarrow{u} D \xleftarrow{\mathbb{d}_i} \mathcal{D}_V(i)$ for each $i \in \mathbb{I}$ and u is monic.

Proof. First assume that the colimit $\mathbb{d}$ is preserved by the graphing functor $\Gamma_{\mathscr{M}} : \mathbb{C} \to \mathrm{Par}(\mathbb{C})$. Then apply Lemma B.3 to infer that $\mathbb{d}$ is universal, i.e. the implication (6)⇒(a) holds. The converse, i.e. that (a) implies (a), is true as well by Lemma B.2.

Conversely, assume that (a) and (6) are equivalent. As the implication (6)⇒(a) holds, the colimit $\mathbb{d}$ is universal; however then it is also partial Van Kampen by Lemma B.2. □

B.1 Coprojections in partial Van Kampen colimits

One of the characteristic properties of adhesive categories is that pushouts preserve monomorphisms; Further, the definition of weak adhesive HLR actually makes this requirement part of the definition. The remainder of this section is devoted to the generalization of this result to arbitrary diagrams of meet semi-lattice shape.

Definition B.5 (Meet semi-lattices) Let $\langle P, \leq \rangle$ be a partially ordered set. Then $\langle P, \leq \rangle$ is a *meet semi-lattice* if for every two elements $p, q \in P$ there exists a greatest lower bound $p \wedge q \in P$.

The next lemma generalizes Lemma 2.3 of [Lack and Sobociński, 2005] to colimits the shapes of which are meet semi-lattices. To make the presentation closer to [Cockett and Guo, 2007], diagrams are taken to be functors from an index category $\mathbb{I}$. When reasoning about these (functor-)diagrams of shape $\mathbb{I}$, objects $i \in \mathbb{I}$ and arrows $x : i \to j$ in $\mathbb{I}$ will often be referred to as *vertices* and *arcs*, respectively.

Further recall that a category $\mathbb{I}$ is a poset if for each pair of objects $i, j \in \mathbb{I}$ the homset $\mathbb{I}(i, j)$ contains at most one arrow; then $i \leq j$ holds true by definition if $\mathbb{I}(i, j)$ is non-empty. Moreover, if a category $\mathbb{I}$ is a poset then each diagram of shape $\mathbb{I}$ commutes, and finally $\mathbb{I}$ is a meet semi-lattice if and only if $\mathbb{I}$ has binary products.

Lemma B.6 (Coprojections in PVK-colimits) Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$; further let $\mathbb{I}$ be a category with binary products, i.e. $\mathbb{I}$ is a meet semi-lattice. Let $\mathcal{D} : \mathbb{I} \to \mathbb{C}$ be a diagram and let $\mathbb{d} : \mathcal{D} \to \Delta_D^{\mathbb{I}}$ be an $\mathscr{M}$-partial Van Kampen colimit; finally let $i \in \mathbb{I}$ be a vertex.

Then the coprojection $\mathbb{d}_i : \mathcal{D}_V(i) \to D$ is an $\mathscr{M}$-morphism provided that only arrows "below" i do *not* belong to $\mathscr{M}$, i.e. provided that for any $j \xrightarrow{x} \ell$ in $\mathbb{I}$ the $\mathbb{C}$-morphism $\mathcal{D}_V(j) \xrightarrow{\mathcal{D}_E(x)} \mathcal{D}_V(\ell)$ is an $\mathscr{M}$-morphism unless $\ell \leq i$ holds true.

Proof. Below will be constructed a suitable diagram $\mathcal{E}\colon \mathbb{I} \to \mathbb{C}$, and a cartesian diagram morphism $\mathbb{m}\colon \mathcal{E} \hookleftarrow \mathcal{D}$ consisting of $\mathscr{M}$-morphisms such that the colimit $\mathbb{e}$ of the diagram $\mathcal{E}$ has $\Delta^{\mathbb{I}}_{\mathcal{D}_V(i)}$ as the codomain, and moreover the coprojection $\mathbb{d}_i\colon \mathcal{D}_V(i) \to D$ is the unique mediating morphism from the colimit $\mathbb{e}$ to $\mathbb{d} \circ \mathbb{m}$.

$$\begin{array}{ccc} \mathcal{E} & \xrightarrow{\mathbb{e}} & \Delta_{\mathcal{D}i} \\ {\scriptstyle \mathbb{m}}\big\downarrow & & \big\downarrow{\scriptstyle \Delta_{\mathbb{d}_i}} \\ \mathcal{D} & \xrightarrow[\mathbb{d}]{} & \Delta_D \end{array}$$

As a rough description, the diagram $\mathcal{E}$ coincides with $\mathcal{D}$ on the region "below" i, and consists of identities everywhere else. Further the diagram morphism $\mathbb{m}$ consists of identities on the region "below" i, and on any vertex k outside this region, the component $\mathbb{m}_k$ is defined as the "closest" $\mathscr{M}$-morphism originating from the region "below" i and leading to k.

$$\begin{array}{ccc} \mathcal{E}j & \xrightarrow{\mathcal{E}x} & \mathcal{E}\ell \\ {\scriptstyle \mathbb{m}_j}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle \mathbb{m}_\ell} \\ \mathcal{D}j & \xrightarrow[\mathcal{D}x]{} & \mathcal{D}\ell \end{array}$$

First let $\lfloor i \rfloor = \{\ell \in \mathbb{I} \mid \ell \leq i\}$ be the collection of all vertices "below" the vertex i from the premises. Next define $\mathcal{E}_V(j) := \mathcal{D}_V(j)$ for each $j \in \lfloor i \rfloor$, and further $\mathcal{E}_E(x) := \mathcal{D}_E(x)$ for each arc $x\colon j \to \ell$ in $\mathbb{I}$ such that $\ell \in \lfloor i \rfloor$ (and hence also $j \in \lfloor i \rfloor$). Moreover define $\mathbb{m}_j := \mathrm{id}_{\mathcal{D}(j)}$ for each $j \in \lfloor i \rfloor$; in this way, naturality pullback squares are obtained for each arc $x\colon j \to \ell$ with $j, \ell \in \lfloor i \rfloor$.

Let $k \in \mathrm{ob}(\mathbb{I}) \setminus \lfloor i \rfloor$ be one of the remaining vertices, and let $k \times i$ be the product of k and i in $\mathbb{I}$, i.e. the greatest lower bound of k and i. As $(k \times i) \leq i$ trivially holds true, the vertex $k \times i$ is contained in the collection $\lfloor i \rfloor$, and hence $\mathcal{E}_V(k \times i)$ is already defined. Now define $\mathcal{E}_V(k) := \mathcal{E}_V(k \times i)$ and $\mathbb{m}_k := \mathcal{D}_E(x^k_{k\times i})$ where $x^k_{k\times i}\colon k \times i \to k$ is the unique (projection) arc from $k \times i$ to k in $\mathbb{I}$.

$$\begin{array}{ccc} \mathcal{E}j & \xlongequal{\mathcal{E}x} & \mathcal{E}\ell \\ {\scriptstyle \mathbb{m}_j}\big\| & \lrcorner & \big\uparrow{\scriptstyle \mathbb{m}_\ell \mathcal{D}x} \\ \mathcal{D}j & \xhookrightarrow[\mathcal{D}x]{} & \mathcal{D}\ell \end{array}$$

To complete the definition of the diagram $\mathcal{E}$ on the remaining arcs, set $\mathcal{E}_E(x) := \mathrm{id}_{\mathcal{E}(\ell)}$, for each arc $x\colon j \to \ell$ in $\mathbb{I}$ with $\ell \notin \lfloor i \rfloor$; also note that $\ell \notin \lfloor i \rfloor$ implies $j \notin \lfloor i \rfloor$ and hence $\mathcal{E}$ is properly defined. Moreover, in this way for each arc $x\colon j \to \ell$ in $\mathbb{I}$ with $\ell \notin \lfloor i \rfloor$, i.e. $\ell \not\leq i$, a naturality pullback square arises from the pullback of $\mathcal{D}_E(x)$ along itself since $\mathcal{D}_E(x)$ is an $\mathscr{M}$-morphism by assumption.

Finally the family $\mathbb{e} := \{\mathcal{E}_E(x^i_{h\times i})\colon \mathcal{E}_V(h) \to \mathcal{D}_V(i)\}_{h\in\mathbb{I}}$ is a colimit for $\mathcal{E}$ where the arc $x^i_{h\times i}\colon h \times i \to i$ is the projection arc in $\mathbb{I}$. As the morphism $\mathbb{d}_i\colon \mathcal{D}_V(i) \to D$ is easily checked to be a mediating morphism from $\mathbb{e}$ to $\mathbb{d} \circ \mathbb{m}$ and since $\mathbb{m}\colon \mathcal{E} \hookleftarrow \mathcal{D}$ is a cartesian diagram morphism which only consists of $\mathscr{M}$-morphisms, the morphism $\mathbb{d}_i$ actually belongs to the class $\mathscr{M}$ by Theorem B.4. □

B.2 Adding universality

So far, colimits have only been hereditary. Now, assuming also universality, some relevant additional properties can be derived.

Definition B.7 (Pullback compatibility) Let $\mathbb{C}$ be a category, let I be a graph, let $\mathcal{D}\colon I \to \mathbb{C}$ be a diagram; further let $\mathbb{d}\colon \mathcal{D} \to \Delta^I_D$ be a colimit, and let $\mathbb{c}\colon \mathcal{D} \to \Delta^I_C$ be a cocone. Finally let $\mathbb{C}$ have enough pullbacks, i.e. let $\mathcal{D}_V(i) \leftarrow_{y_{ij}} X_{ij} \rightarrow_{z_{ij}} \mathcal{D}_V(j)$ be the pullback of $\mathcal{D}_V(i) \rightarrow_{\mathbb{d}_j} D \leftarrow_{\mathbb{d}_j} \mathcal{D}_V(j)$ for each pair $i, j \in V_I$.

Then the cocone $\mathbb{c}$ is *pullback compatible with* the colimit $\mathbb{d}$, if for each pair $i, j \in V_I$ the span $\mathcal{D}_V(i) \leftarrow_{y_{ij}} X_{ij} \rightarrow_{z_{ij}} \mathcal{D}_V(j)$ is also a pullback of the cospan $\mathcal{D}_V(i) \rightarrow_{\mathbb{c}_j} C \leftarrow_{\mathbb{c}_j} \mathcal{D}_V(j)$.

The main examples of pullback compatible cocones are those which consist of monomorphisms only; Nevertheless, the following more general result holds, which is used to Corollary , which in turn can be used to obtain unions in weakly (ω-)adhesive categories.

Lemma B.8 (Monic gaps from universality) Let I be a graph. Let $\mathbb{C}$ be a category with (enough[87]) pullbacks, let $\mathcal{D}\colon I \to \mathbb{C}$ be a diagram with colimit $\mathbb{d}\colon \mathcal{D} \to \Delta^I_D$. Further let $\mathbb{c}\colon \mathcal{D} \to \Delta^I_Z$ be a cocone that is pullback compatible with $\mathbb{d}$.

Then the unique mediating morphism $u\colon \mathbb{d} \to \mathbb{c}$ is a monomorphisms if colimits of shape I are universal in $\mathbb{C}$.

Proof. The claim is that the mediating morphism $u\colon \mathbb{d} \to \mathbb{c}$ is monic. Hence let $f, g\colon E \to D$ in $\mathbb{C}$ such that $u \circ f = u \circ g$. Now construct the following two pullbacks in the category $\mathbb{Diag}[I, \mathbb{C}]$: first let $\Delta_E \leftarrow \prec \mathcal{F} \rightarrow \mathcal{D}$ be the pullback of $\Delta_E \rightarrow_{\Delta_f} \Delta_D \leftarrow_{\mathbb{d}} \mathcal{D}$, yielding the pullback square $\begin{smallmatrix}\Delta_E & \leftarrow \\ \Delta_D & \leftarrow\end{smallmatrix} \downarrow \begin{smallmatrix}\lrcorner & \mathcal{F} \\ & \mathcal{D}\end{smallmatrix}$, and second let $\Delta_E \leftarrow \prec \mathcal{G} \rightarrow \mathcal{D}$ be the pullback of $\Delta_E \rightarrow_{\Delta_g} \Delta_D \leftarrow_{\mathbb{d}} \mathcal{D}$, resulting in another pullback square $\begin{smallmatrix}\Delta_E & \leftarrow \\ \Delta_D & \leftarrow\end{smallmatrix} \downarrow \begin{smallmatrix}\lrcorner & \mathcal{G} \\ & \mathcal{D}\end{smallmatrix}$ (see the left diagram in Figure 100). As the colimit $\mathbb{d}\colon \mathcal{D} \to \Delta^I_D$ is universal, also the two cocones $\mathbb{f}\colon \mathcal{F} \to \Delta^I_E$ and $\mathbb{g}\colon \mathcal{G} \to \Delta^I_E$ are colimits; this implies that the family $\{\mathbb{f}_i\colon \mathcal{F}_V(i) \to E\}_{i \in V_I}$ is jointly epic in $\mathbb{C}$.

Further, for each node $i \in V_I$, pull the colimit $\mathbb{g}\colon \mathcal{G} \to \Delta^I_E$ back along the i-component $\mathbb{f}_i\colon \mathcal{F}_V(i) \to E$, i.e. let $\Delta^I_{\mathcal{F}_V(i)} \leftarrow_{\mathbb{k}_i} \mathcal{H}_i \rightarrow_{\mathbb{l}_i} \Delta^I_{\mathcal{G}_V(i)}$ be the pullback of $\Delta^I_{\mathcal{F}_V(i)} \rightarrow_{\mathbb{f}_i} \Delta^I_E \leftarrow_{\mathbb{g}} \Delta^I_{\mathcal{G}_V(i)}$, which is the construction corresponding to the top face of Figure 100. As $\mathbb{g}\colon \mathcal{G} \to \Delta^I_E$ is a colimit of shape I, and colimits of shape I are universal by assumption, pulling back along $\mathbb{f}_i\colon \mathcal{F}_V(i) \to E$ yields a colimit

[87] Below, the present lemma is used in a case where the category $\mathbb{C}$ only has pullbacks along monomorphisms.

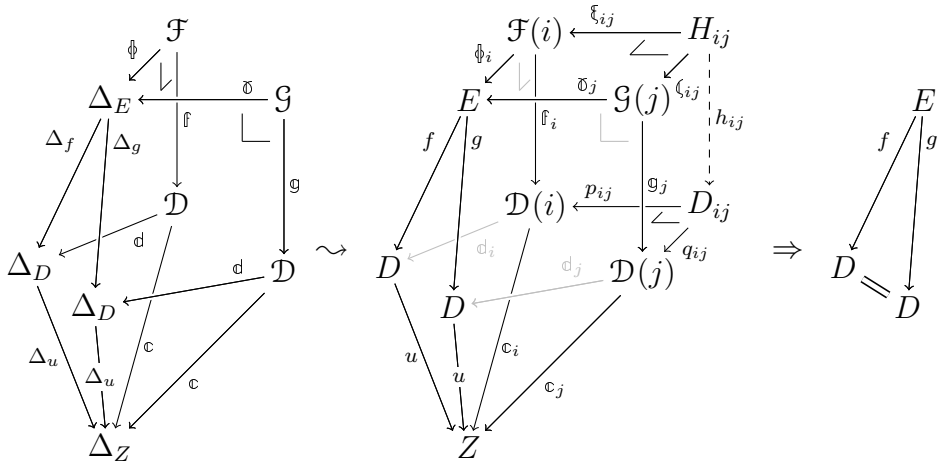


Figure 100: Proof sketch of Proposition B.8

$\mathfrak{k}_i \colon \mathcal{H}_i \to \Delta^I_{\mathcal{F}_V(i)}$; this implies that the family $\{\mathfrak{k}_{ij} \colon \mathcal{H}_{iV}(j) \to \mathcal{F}_V(i)\}_{j \in V_I}$ is jointly epic for each $i \in V_I$.

As a final construction, for each $i, j \in V_I$, let $\mathcal{D}_V(i) \leftarrow p_{ij} - D_{ij} - q_{ij} \rightarrow \mathcal{D}_V(j)$ be a pullback of $\mathcal{D}_V(i) - \mathbb{d}_i \rightarrow D \leftarrow \mathbb{d}_j - \mathcal{D}_V(j)$, i.e.

$$\mathbb{d}_i \circ p_{ij} = \mathbb{d}_j \circ q_{ij} \qquad \text{for all } i, j \in V_I. \tag{16}$$

Further the span $\mathcal{D}_V(i) \leftarrow p_{ij} - D_{ij} - q_{ij} \rightarrow \mathcal{D}_V(j)$ is also a pullback of the cospan $\mathcal{D}_V(i) - \mathbb{c}_i \rightarrow Z \leftarrow \mathbb{c}_j - \mathcal{D}_V(j)$ since the cocone $\mathbb{c}$ is pullback compatible with the colimit $\mathbb{d}$ by assumption. These constructions imply that there is a mediating morphism $h_{ij} \colon \mathcal{H}_{iV}(j) \to D_{ij}$ as shown in the middle diagram of Figure 100; the necessary equations are the following.

$$\begin{aligned}
\underline{\mathbb{c}_i} \circ \mathbb{f}_i \circ \mathfrak{k}_{ij} &= u \circ \underline{\mathbb{d}_i \circ \mathbb{f}_i} \circ \mathfrak{k}_{ij} && \text{(as } u \colon \mathbb{d} \to \mathbb{c}\text{)} \\
&= \underline{u \circ f} \circ \mathfrak{f}_i \circ \mathfrak{k}_{ij} && \text{(by construction of } \mathfrak{f}\text{)} \\
&= u \circ g \circ \underline{\mathfrak{f}_i \circ \mathfrak{k}_{ij}} && \text{(by assumption about } f \text{ and } g\text{)} \\
&= u \circ \underline{g \circ \mathfrak{G}_j} \circ \mathfrak{h}_{ij} && \text{(by construction of } \mathfrak{k}_i\text{)} \\
&= \underline{u \circ \mathbb{d}_j} \circ \mathfrak{g}_j \circ \mathfrak{h}_{ij} && \text{(by construction of } \mathfrak{g}\text{)} \\
&= \mathbb{c}_j \circ \mathfrak{g}_j \circ \mathfrak{h}_{ij} && \text{(as } u \colon \mathbb{d} \to \mathbb{c}\text{)}
\end{aligned}$$

This implies that there exist $h_{ij} \colon \mathcal{H}_{iV}(j) \to D_{ij}$ such that

$$\mathbb{f}_i \circ \mathfrak{k}_{ij} = p_{ij} \circ h_{ij} \quad \text{and} \quad \mathfrak{g}_j \circ \mathfrak{h}_{ij} = q_{ij} \circ h_{ij} \tag{17}$$

holds for each pair $i, j \in V_I$, since the span $\mathcal{D}_V(i) \leftarrow p_{ij} - D_{ij} - q_{ij} \rightarrow \mathcal{D}_V(j)$ is a pullback of the cospan $\mathcal{D}_V(i) - \mathbb{c}_i \rightarrow Z \leftarrow \mathbb{c}_j - \mathcal{D}_V(j)$.

The last step of the proof consist in deriving the equation $f \circ \Phi_i \circ \mathfrak{k}_{ij} = g \circ \Phi_i \circ \mathfrak{k}_{ij}$ for each $i, j \in V_I$, as then the equation $f \circ \Phi_i = g \circ \Phi_i$ must hold for each $i \in V_I$ since the family $\{\mathfrak{k}_{ij} \colon \mathcal{H}_{i_V}(j) \to \mathcal{F}_V(i)\}_{j \in V_I}$ is jointly epic; then the desired equation $f = g$ follows because the family $\{\Phi_i \colon \mathcal{F}_V(i) \to E\}_{i \in V_I}$ is jointly epic.

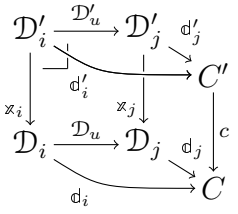
$$
\begin{aligned}
f \circ \Phi_i \circ \mathfrak{k}_{ij} &= \mathbb{d}_i \circ \mathbb{f}_i \circ \mathfrak{k}_{ij} && \text{(by construction of } \mathbb{f}) \\
&= \mathbb{d}_i \circ p_{ij} \circ h_{ij} && \text{(by Equation (17))} \\
&= \mathbb{d}_j \circ q_{ij} \circ h_{ij} && \text{(by Equation (16))} \\
&= \mathbb{d}_j \circ \mathbb{g}_j \circ \mathfrak{c}_{ij} && \text{(by Equation (17))} \\
&= g \circ \mathfrak{G}_j \circ \mathfrak{c}_{ij} && \text{(by construction of } \mathfrak{G}) \\
&= g \circ \Phi_i \circ \mathfrak{k}_{ij} && \text{(by construction of } \mathfrak{k}_i)
\end{aligned}
$$

□

Van Kampen colimits from universal partial ones. One fact, which has not been used in this thesis but might be related to [Cockett and Guo, 2007] is that universal, hereditary colimits of diagrams that consist only of $\mathscr{M}$-morphisms actually are proper Van Kampen colimits.

☼ Definition B.9 (Van Kampen colimit) Let $\mathcal{D} \colon \mathbb{J} \to \mathbb{C}$ be a diagram and let $\mathbb{d} \colon \mathcal{D} \to \Delta_C^{\mathbb{J}}$ be a colimit.

Then $\mathbb{d}$ is *Van Kampen* (VK) if given a functor $\mathcal{D}' \colon \mathbb{J} \to \mathbb{C}$, a cartesian natural transformation $\mathbb{x} \colon \mathcal{D}' \to \mathcal{D}$, an object $C' \in \mathbb{C}$, a cocone $\mathbb{d}' \colon \mathcal{D}' \to \Delta'_C$ and an arrow $c \colon C' \to C$ such that $c \circ \mathbb{d}'_i = \mathbb{d}_i \circ \mathbb{x}_i$ holds for all $i \in \mathbb{J}$ (the diagram to the right is commutative for all $u \colon i \to j$ in $\mathbb{J}$), the following two conditions are equivalent:



(i) the pair $\langle C', \mathbb{d}' \colon \mathcal{D}' \to \Delta'_C \rangle$ is a colimit;

(ii) for all i, the pair $\mathcal{D}_i \leftarrow \mathbb{x}_i - \mathcal{D}'_i - \mathbb{d}'_i \rightarrow C'$ is a pullback of $\mathcal{D}_i - \mathbb{d}_i \rightarrow C \leftarrow c - C'$.

The implication (ii) ⇒ (i) is universality and the implication in the reverse direction (i) ⇒ (ii) will be referred to as *converse universality*. Thus, with this terminology, a colimit $\mathbb{d}$ is a VK-colimit if and only if it satisfies universality and converse universality. ☼

Example B.10 The following well-known concepts are examples of VK-colimits:

(i) a strict initial object is a VK-colimit for the functor from the empty category;

(ii) a coproduct diagram which satisfies the axioms required of coproducts in extensive categories [Carboni et al., 1993] is VK-colimit for a functor from the discrete two object category;

(iii) a VK-square is a functor from $\cdot \leftarrow \cdot \rightarrow \cdot$ with a VK-colimit. An adhesive category has pushouts along monomorphisms and such pushouts are VK.

Theorem B.11 (VK-colimits from universal PVK-colimits) Let $\mathbb{I}$ be a category that is a poset and has binary products, i.e. $\mathbb{I}$ is a meet semi-lattice. Let $\mathbb{C}$ be a category with an admissible class of monomorphisms $\mathscr{M}$ that has universal, $\mathscr{M}$-partial VK colimits for $\mathscr{M}$-morphism creating functors of shape $\mathbb{I}$, i.e. for any diagram $\mathcal{D} : \mathbb{I} \to \mathbb{D}$ such that $\mathcal{D}_E(x) \in \mathscr{M}$ holds for all $x : j \to \ell$ in $\mathbb{I}$, there exists a colimit $\mathcal{D}$.

Then $\mathbb{C}$-colimits of $\mathscr{M}$-morphism creating functors of shape $\mathbb{I}$ are actually Van Kampen colimits.

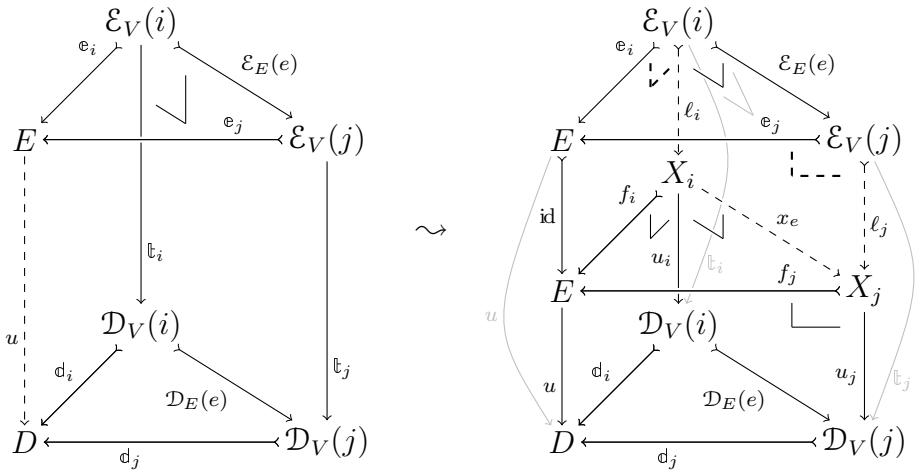
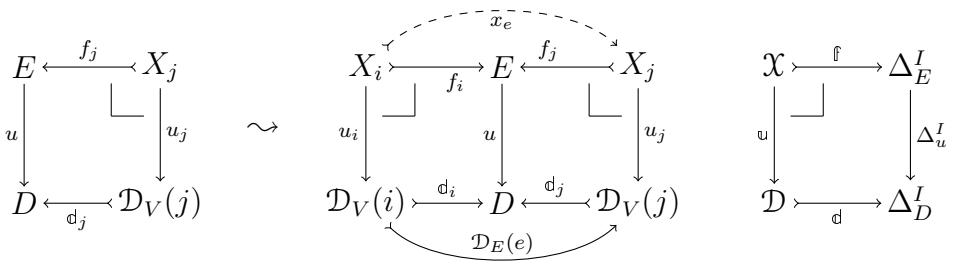


Figure 101: Illustration of the proof of Theorem B.11

Proof. Let $\mathbb{d}\colon \mathcal{D} \to \Delta_D^{\mathbb{I}}$ be the colimit of the diagram $\mathcal{D}$; as the colimit $\mathbb{d}$ is universal by assumption, it is enough to show the "opposite" of universality of $\mathbb{d}$ (see Figure 101). In detail, the claim to be proved is that for each cartesian diagram morphism $\mathbb{t}\colon \mathcal{E} \to \mathcal{D}$ and colimit $\mathbb{e}\colon \mathcal{E} \to \Delta_E^{\mathbb{I}}$ with mediating morphism $u\colon \mathbb{e} \to \mathbb{d}$, the span $E \xleftarrow{\mathbb{e}_j} \mathcal{E}_V(j) \xrightarrow{\mathbb{t}_j} \mathcal{D}_V(j)$ is a pullback of $E \xrightarrow{u} D \xleftarrow{\mathbb{d}_j} \mathcal{D}_V(j)$ for each $j \in \mathbb{I}$.

As the functor $\mathcal{D}\colon \mathbb{I} \to \mathbb{C}$ preserves monos and $\mathbb{I}$ is a poset, the $\mathbb{C}$-arrow $\mathcal{D}_E(e)\colon \mathcal{D}_V(i) \rightarrowtail \mathcal{D}_V(j)$ is monic for each arc $e\colon i \to j$ in $\mathbb{I}$; moreover $\mathbb{I}$ has binaray products and hence Lemma B.6 is applicable and implies that the colimit $\mathbb{d}\colon \mathcal{D} \to \Delta_D^{\mathbb{I}}$ consists of monomorphisms. Now let $\mathbb{t}\colon \mathcal{E} \to \mathcal{D}$ be a cartesian diagram morphism where $\mathcal{E}\colon \mathbb{I} \to \mathbb{C}$ is another $\mathbb{I}$-shaped diagram. This means that also all arrows of the diagram $\mathcal{E}$ are monomorphisms, as monomorphisms are stable under pullback. Hence let $\mathbb{e}\colon \mathcal{E} \to \Delta_D^{\mathbb{I}}$ be the colimit of $\mathcal{E}$, and apply Lemma B.6 once more to deduce that this colimit $\mathbb{e}$ consist of monomorphisms as well. Finally let $u\colon \mathbb{e} \to \mathbb{d}$ be the uniquely determined mediating morphism. The present situation is illustrated in the left hand diagram of Figure 101.

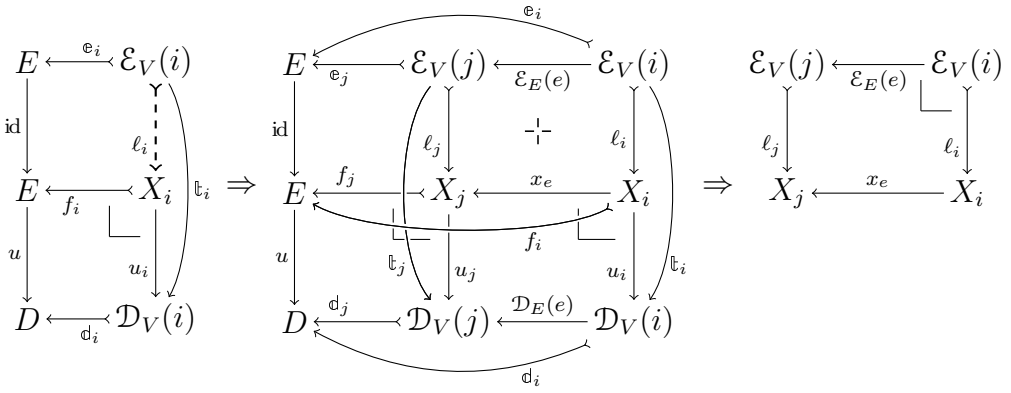
Now the proof idea is to compare the colimit $\mathbb{e}$ with the colimit $(\Delta_u^{\mathbb{I}})^*(\mathbb{d})$ resulting from pulling pulling back $\mathbb{d}$ along u (recall that $\mathbb{d}$ is univeral by assumption); in fact the two colimits $\mathbb{e}$ and $(\Delta_u^{\mathbb{I}})^*(\mathbb{d})$ will turn out to be essentially the same. Hence construct the diagram $\mathcal{X}\colon \mathbb{I} \to \mathbb{C}$ by pulling back $\mathbb{d}\colon \mathcal{D} \to \Delta_D^{\mathbb{I}}$ along u as shown in the following display.



Since $\mathbb{d}$ is universal, the cocone $\mathbb{f}\colon \mathcal{X} \to \Delta_E^{\mathbb{I}}$ is actually a colimit; moreover $\mathbb{u}\colon \mathcal{X} \to \mathcal{D}$ is a cartesian transformation.

The final step of the proof consists in defining a cartesian diagram isomorphism $\mathbb{l}\colon \mathcal{E} \xrightarrow{\cong} \mathcal{X}$ such that the equations $\mathbb{t} = \mathbb{u} \circ \mathbb{l}$ and $\mathbb{e} = \mathbb{f} \circ \mathbb{l}$ hold. Define the family $\mathbb{l} := \{\ell_i\colon \mathcal{E}_V(i) \to \mathcal{X}_V(i)\}_{i \in \mathbb{I}}$ as indicated in the left hand diagram

in the following display.



However the family $\mathbb{l}$ is a actually a monic cartesian diagram morphism $\mathbb{l}\colon \mathcal{E} \rightarrowtail \mathcal{X}$. The proof of this is sketched in the preceeding display; the details for every arc $e\colon i \to j$ in $\mathbb{I}$ are as follows.

$$
\begin{aligned}
\underline{f_j \circ x_e} \circ \ell_i &= f_i \circ \ell_i && \text{(by construction of } x_e\text{)} \\
&= \mathbb{e}_i && \text{(by construction of } \ell_i\text{)} \\
&= \underline{\mathbb{e}_j} \circ \mathcal{E}_E(e) && \text{(as } \mathbb{e} \text{ is a cocone)} \\
&= f_j \circ \ell_j \circ \mathcal{E}_E(e) && \text{(by construction of } \ell_j\text{)} \\
\Rightarrow\ x_e \circ \ell_i &= \ell_j \circ \mathcal{E}_E(e) && \text{(as } f_j \text{ is monic)}
\end{aligned}
$$

The above equations show that $\mathbb{l}\colon \mathcal{E} \rightarrowtail \mathcal{X}$ is a diagram morphism; further it is also cartesian since $\mathbb{t} = \mathbb{u} \circ \mathbb{l}$ holds and both $\mathbb{t}$ and $\mathbb{u}$ are cartesian.

Finally apply Theorem B.4 to the cartesian transformation $\mathbb{l}\colon \mathcal{E} \rightarrowtail \mathcal{X}$, and the two colimits $\mathbb{f}\colon \mathcal{X} \to \Delta_E^{\mathbb{I}}$ and $\mathbb{e}\colon \mathcal{E} \to \Delta_E^{\mathbb{I}}$ with their mediating morphism $\mathrm{id}_E\colon \mathbb{e} \to \mathbb{f} \circ \mathbb{l}$. Hence, for each $i \in \mathbb{I}$, the span $E \leftarrow \mathbb{e}_i - \mathcal{E}_V(i) \rightarrowtail \ell_i \to X_i$ is the pullback of the cospan $E \xrightarrow{\mathrm{id}} E \leftarrow f_i - X_i$, giving rise to the pullback square $\begin{smallmatrix} E & \leftarrow & \mathcal{E}(i) \\ \downarrow & \lrcorner & \downarrow \\ E & \leftarrow & X_i \end{smallmatrix}$ (see the right hand diagram in Figure 101). □

Trace processes and switch-equivalence

C.1 Static characterization of trace processes

C

The trace process construction of Definition 4.8 gives rise to a functor from the category of derivations to the category of oriented transformation systems.

☼ Definition C.1 (Category of oriented transformation systems) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category. The *category of oriented transformation systems in* $\mathbb{C}$, written $\textsc{ots}(\mathbb{C})$, has the same objects as $\mathbb{C}$. To define the auxiliary notion of proto-morphism, let $A, B \in \mathbb{C}$ be objects; now, a *proto-morphism* from A to B is an oriented transformation system $\langle s, Q, e\rangle$ with $\operatorname{dm} s = A$ and $\operatorname{dm} e = B$; the fact that $\langle s, Q, e\rangle$ is a proto-morphism from A to B is also expressed by writing $\langle s, Q, e\rangle \colon A \rightsquigarrow B$.

A morphism from A to B is defined in terms of this auxiliary notion of proto-morphism as an isomorphism class of proto-morphisms $[\langle s, Q, e\rangle] \colon A \to B$ where $\langle s, Q, e\rangle \colon A \rightsquigarrow B$ is a proto-morphisms from A to B and

$$\big[\langle s, Q, e\rangle\big] = \Big\{ \langle s', Q', e'\rangle \;\Big|\; \langle s, Q, e\rangle \cong \langle s', Q', e'\rangle \Big\}.$$

The identity on an object A is $[\langle \mathrm{id}_A, \varnothing, \mathrm{id}_A\rangle]$, and for the composition of two arrows $[\langle s_1, Q_1, e_1\rangle] \colon A \to B$ over T_1 and $[\langle s_2, Q_2, e_2\rangle] \colon B \to C$ over T_2 take a pushout $T_1 \leftarrow x_1 \rightarrowtail T \leftarrow x_2 \rightarrowtail T_2$ of $T_1 \leftarrow e_1 \rightarrowtail B \leftarrow s_2 \rightarrowtail T_2$ and assemble the two systems over T, i.e.

$$\left[\left\langle \begin{array}{l} s_1, \\ Q_1, \\ e_1 \end{array} \right\rangle\right] \fatsemi \left[\left\langle \begin{array}{l} s_2, \\ Q_2, \\ e_2 \end{array} \right\rangle\right] = \left[\left\langle \begin{array}{l} s_1 \fatsemi x_1, \\ \Sigma_{x_1}(Q_1) \cup \Sigma_{x_2}(Q_2), \\ x_2 \circ e_2 \end{array} \right\rangle\right] \text{ where } \begin{array}{ccc} & & B & & \\ & {\scriptstyle e_1}\swarrow & & \searrow{\scriptstyle s_2} & \\ T_1 & & & & T_2 \,. \\ & {\scriptstyle x_1}\searrow & & \swarrow{\scriptstyle x_2} & \\ & & T & & \end{array}$$

☼

The precise formulation of the fact that the construction of trace process as described in Definition 4.8 gives rise to a functor $\mathcal{P} \colon \textsc{dpo}(\mathbb{C})_{\mathscr{M}} \to \textsc{ots}(\mathbb{C})$ is given by the next definition and proposition.

☼ Definition C.2 (Process construction) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, and let $\mathfrak{X} \colon A \Longrightarrow B$ be a derivation in $\textsc{dpo}(\mathbb{C})_{\mathscr{M}}$.

Then the *process construction* applied to $\mathfrak{X}$ yields the isomorphism class $[\![\mathfrak{X}]\!]$ where $\langle\!\langle \mathfrak{X} \rangle\!\rangle$ is any trace process of $\mathfrak{X}$. This induces a graph morphism $\mathcal{P} \colon \mathcal{U}(\textsc{dpo}(\mathbb{C})_{\mathscr{M}}) \to \mathcal{U}(\textsc{ots}(\mathbb{C}))$ from the underlying graph of the derivation category to the one of the category of OTSs which acts as the identity on the objects of $\mathbb{C}$ and maps sequences of matching direct derivation diagrams to the corresponding isomorphism class of trace processes, i.e. $\mathcal{P} = \langle \mathrm{id}_{\mathrm{ob}(\mathbb{C})}, [\![\text{-}]\!]\rangle$. ☼

◌ Proposition C.3 (Process construction functor) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category. Then the graph morphism $\mathcal{P} = \langle \mathrm{id}_{\mathrm{ob}(\mathbb{C})}, [\![\llcorner\!\lrcorner]\!] \rangle$ as described in Definition C.2 actually is a functor $\mathcal{P}\colon \textsc{dpo}(\mathbb{C})_{\mathscr{M}} \to \textsc{ots}(\mathbb{C})$.

Proof. The main observation is that the type object of a trace processes arises as the colimit of the diagram that is induced by the original derivation as described in [Baldan et al., 2006a]. Moreover pushouts can be used to combine two such colimits. □

The non-full subcategory of $\textsc{ots}(\mathbb{C})$ that lies in the image of the functor $\mathcal{P}\colon \textsc{dpo}(\mathbb{C})_{\mathscr{M}} \to \textsc{ots}(\mathbb{C})$ will be called the category of *constructed processes*, which will turn out to coincide with subcategory of concatenable processes (see Definition 4.16).

☼ Definition C.4 (Category of constructed processes) The category of *constructed processes in* $\mathbb{C}$, written $\textsc{cp}(\mathbb{C})$, is the lluf[88] subcategory of $\textsc{ots}(\mathbb{C})$ which has exactly those isomorphism classes of OTSs as arrows which are in the image of the trace process functor $\mathcal{P}\colon \textsc{dpo}(\mathbb{C})_{\mathscr{M}} \to \textsc{ots}(\mathbb{C})$, i.e. the equation $\mathrm{ob}(\textsc{cp}(\mathbb{C})) = \mathrm{ob}(\textsc{ots}(\mathbb{C}))$ holds and

$$\mathrm{ar}(\textsc{cp}(\mathbb{C})) = \left\{ \xi \in \mathrm{ar}(\textsc{ots}(\mathbb{C})) \;\middle|\; \exists \underset{\sim}{\mathfrak{X}} \in \mathrm{ar}(\textsc{dpo}(\mathbb{C})_{\mathscr{M}}).\, \xi = \mathcal{P}(\underset{\sim}{\mathfrak{X}}) \right\}.$$

☼

The first group of facts about constructed processes that can be proven using the construction of trace processes of Definition 4.8, are *acyclicity*, *causal soundness* and *completeness*.

◌ Proposition C.5 (Properties of constructed processes I) Let $\mathbb{C}$ be any weakly $\mathscr{M}$-adhesive category; let $[\langle s, Q, e\rangle]\colon A \to B$ in $\textsc{cp}(\mathbb{C})$ be a constructed process. Then the representative OTS $\langle s\colon A \hookleftarrow T, Q, e\colon B \hookleftarrow T\rangle$ satisfies the following.

ACYCLICITY The asymmetric conflict relation $\nearrow \subseteq Q \times Q$ is acyclic.

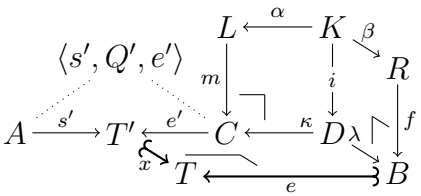
CAUSAL SOUNDNESS The start object has no causes, i.e. $\lfloor s \rfloor = \varnothing$.

CAUSAL COMPLETENESS Every subobject is covered by its causes and the start object; more detailed for every subobject $[w] \in \mathrm{Sub}_{\mathscr{M}}(T)$ the inclusion $w \sqsubseteq [s \sqcap w] \sqcup \bigsqcup_{q \in \lfloor w \rfloor} [r_q \sqcap w]$ holds.

[88] Recall that this means nothing else but having the same collection of objects; in other words, a subcategory $\mathbb{D} \subseteq \mathbb{C}$ is a lluf subcategory if $\mathrm{ob}(\mathbb{D}) = \mathrm{ob}(\mathbb{C})$.

Proof. The proof is by induction on the size of Q. If Q is empty, acyclicity, and causal soundness are trivial; further also causal completeness is immediate, as in the base case the equations $A = B = T$ and $s = \mathrm{id}_A = \mathrm{id}_B = \mathrm{id}_T$ hold since $[\langle s, Q, e\rangle]$ is the identity process on A.

For the induction step consider the construction of trace processes of Definition 4.8 shown to the right; the illustration shows $\langle s, Q, e\rangle$ over T arising from a smaller process $Y' = \langle s', Q', e'\rangle$ over T' by addition of a new rule occurrence of $L \leftarrow\!\alpha\!\rightarrow K \leftarrow\!\beta\!\rightarrow R$ at the match $m\colon L \hookleftarrow C$ yielding the new rule occurrence



$$q = (\underbrace{x \circ e' \circ m}_{l_q}) \leftarrow\!\alpha\!\rightarrow (\underbrace{e \circ \lambda \circ i}_{k_q}) \leftarrow\!\beta\!\rightarrow (\underbrace{e \circ f}_{r_q}),$$

i.e. $Q = \{q\} \cup \Sigma_x(Q')$ and $s = x \circ s'$.

As for acyclicity, observe that for each rule $q' \in \Sigma_x(Q')$ the inclusion $l_{q'} \sqsubseteq x$ holds for the left hand side $l_{q'}$ of q'. As consequence, $q \not\prec q'$ holds for all $q' \in \Sigma_x(Q')$ since $r_q \sqcap x \sqsubseteq k_q$. Further, using causal soundness for the opposite $[Y']^\circ$ of the constructed process $[Y']$ where $[Y']^\circ = [\langle e', Q'^\circ, s'\rangle]\colon C \to A$, conclude that $l_{\bar{q}} \sqcap e' \sqsubseteq k_{\bar{q}}$ must hold for all rules $(l_{\bar{q}} \leftarrow k_{\bar{q}} \rightarrow r_{\bar{q}}) \in Q'$ of Y'. This together with the inclusion $e' \circ m \sqsubseteq e'$ implies that also $l_{\bar{q}} \sqcap [e' \circ m] \sqsubseteq k_{\bar{q}}$ holds, i.e. $l_{q'} \sqcap l_q \sqsubseteq k_{q'}$ is true for all $q' \in \Sigma_x(Q')$. Summarizing, this paragraph yields $q \not\ll q'$ and $q \not\prec q'$ for all $q' \in \Sigma_x(Q')$. By the induction hypothesis, $\langle s', Q', e'\rangle$ is acyclic, and hence every linearization of $\nearrow_{Q'}$ induces one of $\nearrow_{\Sigma_x(Q')}$ which then can be extended with q; in other words, also $\nearrow_Q$ is acyclic.

Causal soundness follows from $r_q \sqcap x = k_q \sqsubseteq k_q$, the inclusion $s \sqsubseteq x$ and the induction hypothesis; to show causal completeness, let $[w] \in \mathrm{Sub}_{\not\prec}(T)$. By construction of $\langle s, Q, e\rangle$, the equation $[\mathrm{id}_T] = [x] \sqcup [r_q]$ holds and T arises as a pushout object; by distributivity (Proposition 7.18), the subobject $[w]$ is the join $[w] = [x \sqcap w] \sqcup [w \sqcap r_q]$. The induction hypothesis implies the equation $[w \sqcap x] = [s \sqcap w] \sqcup \bigsqcup_{q \in \lfloor w \sqcap x \rfloor} [r_q \sqcap w]$. Inserting the latter equation into $[w] = [x \sqcap w] \sqcup [w \sqcap r_q]$ yields

$$[w] = \Big([s \sqcap w] \sqcup \bigsqcup_{q \in \lfloor w \sqcap x \rfloor} [r_q \sqcap w]\Big) \sqcup [w \sqcap r_q].$$

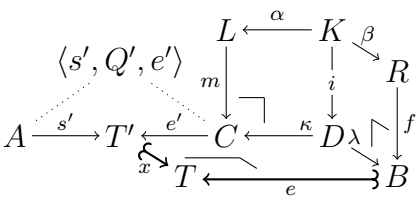
□

In weakly $\mathscr{M}$-adhesive categories with $\mathscr{M}$-effective unions, the three properties of Proposition C.5 and their opposites (see Corollary C.7 below) suffice to characterize those OTSs that arise as the trace process of some derivation. However, effective unions are not necessary, which is ensured by the following lemma, which again has a direct proof using induction on the size of trace processes.

Lemma C.6 (Dissection of trace processes) Let $\mathbb{C}$ be any weakly $\mathscr{M}$-adhesive category, and let $\langle s, Q, e\rangle$ be a trace process over T. Then $\langle s, Q, e\rangle$ has the *dissection property*, i.e. for any $Q' \subseteq Q$ the join $s \sqcup \bigsqcup_{q\in\lfloor Q'\rfloor} r_q$ exists and belongs to $\mathrm{Sub}_{\mathscr{M}}(T)$.

Proof. The proof is by induction of the size of Q and the base case, i.e. $Q = \varnothing$ is trivial.

For the induction step consider the construction of trace processes of Definition 4.8 shown to the right; the illustration shows $\langle s, Q, e\rangle$ over T arising from a smaller process $Y' = \langle s', Q', e'\rangle$ over T' by addition of a new rule occurrence of $L \leftarrow\alpha\rightarrow K \leftarrow\beta\rightarrow R$ at the match $m\colon L \hookrightarrow C$ yielding the new rule occurrence



$$q = (\underbrace{x \circ e' \circ m}_{l_q}) \leftarrow\alpha\rightarrow (\underbrace{e \circ \lambda \circ i}_{k_q}) \leftarrow\beta\rightarrow (\underbrace{e \circ f}_{r_q}),$$

i.e. $Q = \{q\} \cup \Sigma_x(Q')$ and $s = x \circ s'$.

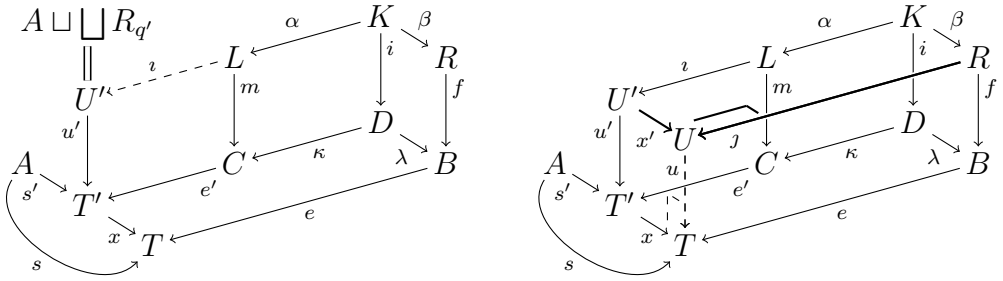


Figure 102: Illustration of the existence of certain joins

To establish the dissection property, let $\bar{Q} \subseteq Q$ be a set such that, w.l.o.g. $\lfloor\bar{Q}\rfloor = \bar{Q}$. Further, if $q \notin \bar{Q}$ then the desired follows from the induction

hypothesis about Y'. Hence consider the case in which $q \in \bar{Q}$ holds. Since $\bar{Q}$ is closed under causes, it also contains the (direct) causes of q. Since $x\colon T' \hookleftarrow T$ is a monomorphism, there is a unique set of rules $Q_x \subseteq Q'$ such that $\Sigma_x(Q_x) = \bar{Q} \setminus \{q\}$,

Applying the induction hypothesis to Y' and this set Q_x yields a subobject $u'\colon U' \hookleftarrow T'$ such that $[x \circ u'] = s \sqcup \bigsqcup_{\bar{q} \in \bar{Q} \setminus \{q\}} r_{\bar{q}}$. From causal completeness of Y of Proposition C.5, the inclusion $l_q \sqsubseteq [x \circ u']$ can be derived, which is witnessed by a unique $\mathscr{M}$-morphism $\imath\colon L \hookleftarrow U'$ (see the left hand diagram in Figure 102) Finally, take a pushout $U' \hookleftarrow x' \rightarrow U \hookleftarrow \jmath \rightarrow R$ of $U' \hookleftarrow \imath \circ \alpha \rightarrow K \hookleftarrow \beta \rightarrow R$ as shown on the right in Figure 102 to obtain a pushout square ${}^{U'}_{U} \downarrow {}^{\leftarrow}_{\leftarrow} \downarrow {}^{K}_{R}$. This yields a mediating morphism $u\colon U \to T$ and a pushout square ${}^{U'}_{T'} \downarrow {}^{\rightarrow}_{\rightarrow} \downarrow {}^{U}_{T}$ by the Pushout Lemma; hence the join $s \sqcup \bigsqcup_{\bar{q} \in \bar{Q}} r_{\bar{q}} = [u]$ exists and belongs to $\mathrm{Sub}_{\mathscr{M}}(T)$. □

Using the fact that every derivation $\mathfrak{X}\colon A \Longmapsto B$ has an opposite derivation $\mathfrak{X}^\circ\colon B \Longmapsto A$ using the opposite rules, Proposition C.5 and Lemma C.6 have the following corollary.

Corollary C.7 (Properties of constructed processes II) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category; let $[\langle s, Q, e\rangle]\colon A \to B$ in $\mathrm{CP}(\mathbb{C})$ be a constructed process. Then the representative OTS $\langle s\colon A \hookleftarrow T, Q, e\colon B \hookleftarrow T\rangle$ satisfies the following.

CO-ACYCLICITY The co-asymmetric conflict relation $\nearrow^\circ \subseteq Q^\circ \times Q^\circ$ is acyclic.

CO-CAUSAL SOUNDNESS The end object has no co-causes, i.e. $\lceil e \rceil = \varnothing$.

CO-CAUSAL COMPLETENESS Each subobject is covered by its co-causes and the end object, i.e. for every $w \in \mathrm{Sub}_{\mathscr{M}}(T)$, the inclusion $[w] \sqsubseteq e \bigsqcup_{q \in \lceil w \rceil} l_q$ holds.

CO-DISSECTION For each subset $Q' \subseteq Q$, the join $e \sqcup \bigsqcup_{q \in \lceil Q' \rceil} l_{q'}$ exists and belongs to $\mathscr{M}$.

In fact the above static properties are sufficient to characterize constructed processes, i.e. constructed and concatenable processes coincide. Hence, the theorem is that the underlying OTS of each concatenable process (Definition 4.16) arises by the trace process construction. The main task is to show that for a given (non-empty) OTS of a concatenable process, it is possible to split the OTS into a singleton trace process of a single direct derivation followed by an OTS which again gives rise to a concatenable process.

Theorem C.8 (Trace processes characterization) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, and $Y = \langle s, Q, e\rangle$ be a OTS such that $[\langle s, Q, e\rangle]\colon A \to B$ is a concatenable process. Then $[\langle s, Q, e\rangle]$ is a constructed process.

Proof. The proof is on the size of $Y = \langle s\colon S \hookleftarrow T, Q, e\colon B \hookleftarrow T\rangle$. The base case with $Q = \varnothing$ and $[Y] = \mathrm{id}_A$ in $\mathrm{OTS}(\mathbb{C})$, follows from (co-)causal completeness, which implies that $s \sqsupseteq \mathrm{id}_T$ (and $e \sqsupseteq \mathrm{id}_T$).

If Q is non-empty, there is some rule $q \in Q$ such that q is minimal w.r.t. asymmetric conflict $\nearrow$, i.e. there does not exist any $\bar{q} \in Q \setminus \{q\}$ such that $\bar{q} \nearrow q$. Then $\{q\}$ is closed under causes and $Q \setminus \{q\}$ is closed under co-causes. Hence the joins $s \sqcup r_q$ and $e \sqcup \bigsqcup_{q' \in Q\setminus\{q\}} l_{q'}$ exist and belongs to $\mathrm{Sub}_{\mathscr{M}}(T)$, which implies that meets distribute over them. Moreover $l_q \sqsubseteq s$ holds by causal completeness, the inclusion $r_q \sqsubseteq e \sqcup \bigsqcup_{q' \in Q\setminus\{q\}} l_{q'}$ follows from co-causal completeness, and $l_q \sqcap (e \sqcup \bigsqcup_{q' \in Q\setminus\{q\}} l_{q'}) = k_q$ is true since q is minimal w.r.t. asymmetric conflict $\nearrow$; furthermore $\mathrm{id}_T = l_q \sqcup (e \sqcup \bigsqcup_{q' \in Q\setminus\{q\}} l_{q'})$, which results in the situation depicted in the left hand diagram in Figure 103. The resulting

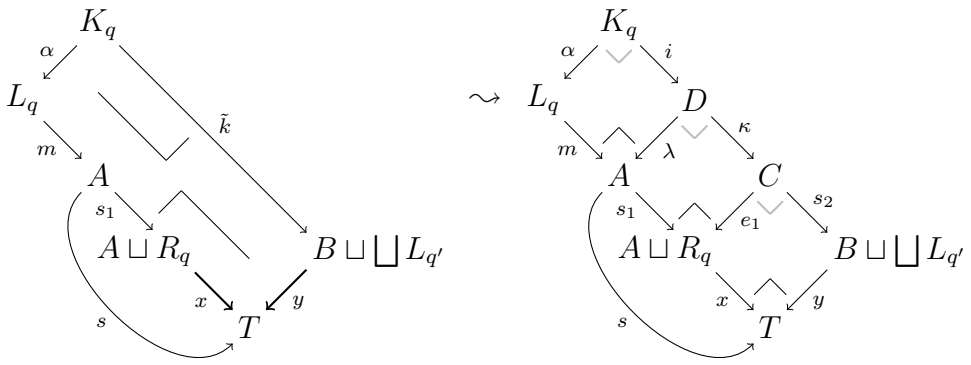


Figure 103: First step of the proof of Theorem C.8

pushout square can be decomposed into three pushout squares using pullbacks as shown on the right in Figure 103. In the following, all arrow names are based on the latter figure (and later also on Figure 104).

The right hand side r_q of q is contained in $[x] = [s \sqcup r_q]$. Further, since q is not a cause of itself, i.e. $r_q \sqcap l_q \sqsubseteq k_q$, also $r_q \sqsubseteq x \circ e_1 = y \circ s_2$ which is witnessed by a unique $\mathscr{M}$-morphism $f\colon R_q \hookleftarrow C$ as shown on the left in Figure 104. Taking a pushout $L_q \hookleftarrow \varphi \twoheadrightarrow V \hookleftarrow \psi \twoheadrightarrow R_q$ of $L_q \hookleftarrow \alpha \twoheadrightarrow K_q \hookleftarrow \beta \twoheadrightarrow R_q$ yields a mediating morphism which witnesses the inclusion $l_q \sqcup r_q \sqsubseteq s \sqcup r_q$ as illustrated. Since the square $\begin{smallmatrix} K & \to \\ A & \to \end{smallmatrix}\!\!\downarrow\!\!\sqcap\!\!\downarrow \begin{smallmatrix} R \\ A \sqcup R_q \end{smallmatrix}$ is actually a pushout square, further pushout squares arise as depicted in the same diagram.

Given the constructions of Figure 104, the left cube diagram corresponds to a singleton trace process. Further there is a unique set of rules Q_2 such

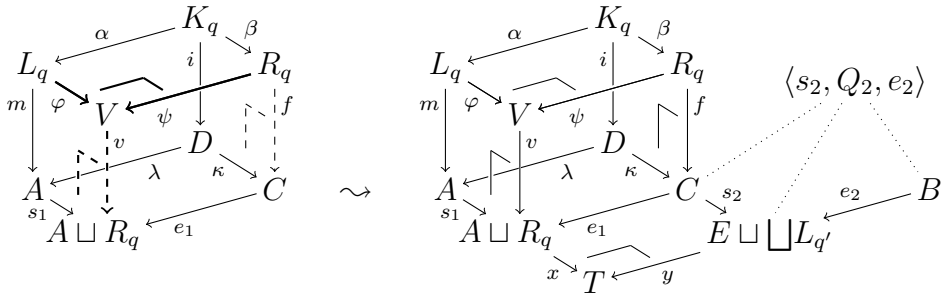


Figure 104: Summary of the constructions in the proof of Theorem C.8

that $\Sigma_y(Q_2) = Q \setminus \{q\}$. It remains[89] to show that the OTS $Y_2 := \langle s_2, Q_2, e_2 \rangle$ actually gives rise to a concatenable process $[Y_2] \colon C \to B$.

Acyclicity and (co-)causal soundness follow immediately, an so does *co*-causal completeness. As for causal completeness, let $w \in \mathrm{Sub}_{\mathscr{M}}(T)$ be e subobject such that let $w \sqsubseteq y$, i.e. $w = y \circ \bar{w}$ for a unique $\mathscr{M}$-morphism $\bar{w}$; now $w \sqcap r_q \sqsubseteq w \sqcap [y \circ s_2] \sqsubseteq [y \circ s_2]$ and $w \sqcap s \sqsubseteq [y \circ s_2 \circ \kappa] \sqsubseteq [y \circ s_2]$ hold, which imply causal completeness of Y_2.

The co-dissection property is inherited directly, and the dissection property follows from distributivity as follows: for every cause closed set of rules $\bar{Q} \subseteq Q_2$ in Y_2 the join $(s \sqcup r_q) \sqcup \bigsqcup_{q' \in \Sigma_y(\bar{Q})} r_{q'}$ exists and is an $\mathscr{M}$-subobject. Using causal completeness of Y, intersection with y yields

$$
\begin{aligned}
y \sqcap \left((s \sqcup r_q) \sqcup \bigsqcup_{\bar{q} \in \bar{Q}} [y \circ r_{\bar{q}}] \right) &= \left((y \sqcap (s \sqcup r_q)) \sqcup \bigsqcup_{\bar{q} \in \bar{Q}} y \sqcap [y \circ r_{\bar{q}}] \right) \\
&= \left([y \circ s_2] \sqcup \bigsqcup_{\bar{q} \in \bar{Q}} [y \circ r_{\bar{q}}] \right) =: [v'].
\end{aligned}
$$

The latter subobject $[v']$ is contained in $[y]$ via a unique $\mathscr{M}$-morphism $v \colon v' \hookleftarrow y$ in $\mathbb{C} \downarrow T$, and $[v] = s_2 \sqcup \bigsqcup_{\bar{q} \in \bar{Q}} r_{\bar{q}}$ is the relevant join in Y_2. □

As becomes clear from the proof of this theorem, by repeated application of the construction in the proof, it is possible to decompose concatenable processes.

[89] The process construction here is "backwards". Note however that the same proof applies to the reversed OTS $Y^\circ = \langle e, Q^\circ, s \rangle \colon B \rightsquigarrow A$.

Scholium C.9 (Process decomposition) Given a concatenable process $[Y] = [\langle s\colon A \to T, Q, e\colon B \to T\rangle]$ with a compatible linearization and $\{q_i\}_{i\in\{\!|\,|Q|\,|\!\}}$ of Q, i.e. $q_{\_}\colon \{\!|\,|Q|\,|\!\} \xrightarrow{\cong} Q$ is a bijective function such that $q_i \nearrow q_j$ implies $i < j$ for all $i, j \in \{\!|\,|Q|\,|\!\}$, then for each $j \in \{\!|\,|Q|\,|\!\}$, there are processes $[Y_j] = [\langle s_j\colon A \to T_j, Q_j, e_j\colon C \to T_j\rangle]$ and $[\bar{Y}]$ such that $[Y] = [Y_j] \mathbin{⨾} [\bar{Y}]$ and $\{q_i \mid i < j\} = \Sigma_x(Q_j)$ for a suitable $\mathscr{M}$-morphism $x\colon T_j \hookrightarrow T$.

C.2 Switching couples via processes

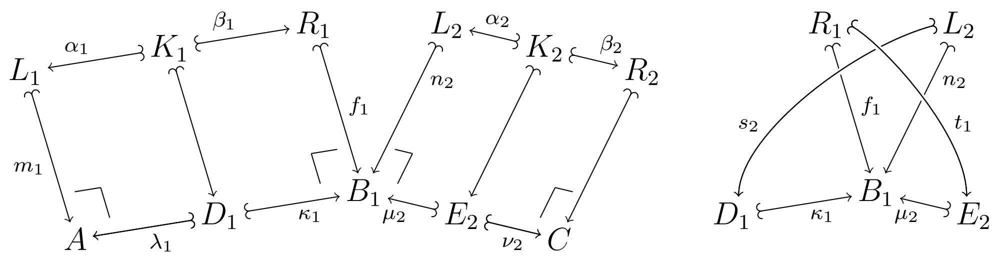


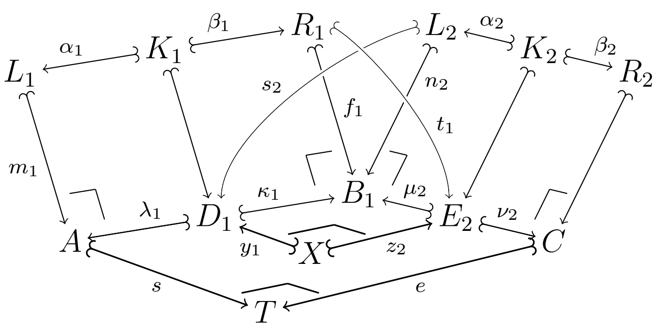
Figure 105: A pair of sequential-independent direct derivation diagrams

Lemma C.10 (Processes of sequential-independent direct derivations) Let $q_1 = L_1 \xleftarrow{\alpha_1} K_1 \xrightarrow{\beta_1} R_1$ and $q_2 = L_2 \xleftarrow{\alpha_2} K_2 \xrightarrow{\beta_2} R_2$ be a pair of $\mathscr{M}$-linear rules, and let $\mathfrak{X}_1\colon A \overset{q_1}{\Rightarrow} B_1$ and $\mathfrak{Y}_2\colon B_1 \overset{q_2}{\Rightarrow} C$ be a pair of sequential-independent direct derivation diagrams in $\mathrm{DPO}(\mathbb{C})_{\mathscr{M}}$ as displayed in Figure 105.

Then (the type object of) a trace process of $\mathfrak{X}_1 \mathbin{⨾} \mathfrak{Y}_2$ can be obtained by taking the pushout of A and C over the intersection of D_1 and E_2, i.e. whenever $\langle A \xrightarrow{s} T, Q, C \xrightarrow{e} T\rangle$ is a trace process of $\mathfrak{X}_1 \mathbin{⨾} \mathfrak{Y}_2$ and $D_1 \xleftarrow{y_1} X \xrightarrow{z_2} E_2$ is a pullback of $D_1 \xrightarrow{\kappa_1} B_1 \xleftarrow{\mu_2} E_2$ then $A \xrightarrow{s} T \xleftarrow{e} C$ is a pushout of $A \xleftarrow{\lambda_1 \circ y_1} X \xrightarrow{z_2 ⨾ \nu_2} C$ (see Figure 106); as an additional property, in the latter situation, $D_1 \xrightarrow{\kappa_1} B_1 \xleftarrow{\mu_2} E_2$ is a pushout of $D_1 \xleftarrow{y_1} X \xrightarrow{z_2} E_2$.

Proof. The proof idea is illustrated in Figure 107. Applying the definition of trace processes (Definition 4.8) yields a diagram as depicted on the left in Figure 107. Taking a pullback $D_1 \xleftarrow{y_1} X \xrightarrow{z_2} E_2$ of $D_1 \xrightarrow{\kappa_1} B_1 \xleftarrow{\mu_2} E_2$ yields a pushout by Lemma 7.8. Applying the pushout lemma successively to the four "bottom" pushouts in Figure 107 results in the fact that $A \xrightarrow{s} T \xleftarrow{e} C$ is a pushout of $A \xleftarrow{\lambda_1 \circ y_1} X \xrightarrow{z_2 ⨾ \nu_2} C$.

□



$$X = D_1 \times_{B_1} E_2$$
$$T = A +_X C$$
$$B_1 = D_1 +_X E_2$$

Figure 106: Trace process of sequential-independent direct derivation diagrams

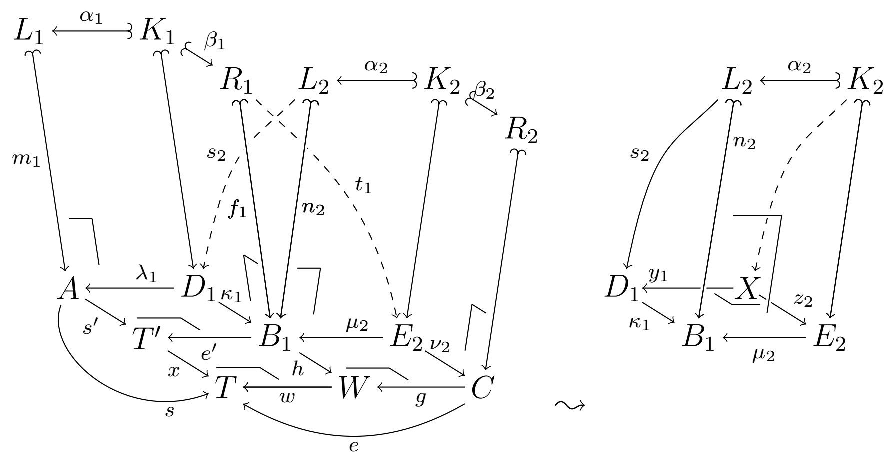


Figure 107: Proof idea of Lemma C.10

The following proposition establishes a direct correspondence between sequential-independence of a pairs of direct derivations diagrams and the causal relations between the corresponding rule occurrences in the induced trace processes.

PROPOSITION C.11 (Sequential independence via trace processes) Let $\mathbb{C}$ be an $\mathscr{M}$-adhesive category, let $q' = L' \leftarrow\alpha'\rightarrowtail K' \leftarrowtail\beta'\rightarrow R'$ and $q = L \leftarrow\alpha\rightarrowtail K \leftarrowtail\beta\rightarrow R$ be two linear rules. Further let $\mathcal{X}\colon A \vDash\langle q',m'\rangle\Rightarrow C$ and $\mathcal{Y}\colon C \vDash\langle q,m\rangle\Rightarrow B$ in $\textsc{dpo}(\mathbb{C},\mathscr{M})_{\mathscr{M}}$ be a pair of direct derivation diagrams which give rise to a trace process $\langle\!\langle \mathcal{X} \mathbin{;} \mathcal{Y} \rangle\!\rangle = \langle A \leftarrowtail s\rightarrow T, \{q'_T, q_T\}, B \leftarrowtail e\rightarrow T\rangle$ of the derivation $\mathcal{X} \mathbin{;} \mathcal{Y}$ where q'_T and q_T are the rule occurrences which result from the application of q' and q, respectively, i.e. $|q'_T| = q'$ and $|q_T| = q$.

Then $\mathcal{X}$ and $\mathcal{Y}$ are sequential-independent if and only if neither $q'_T \prec q_T$

nor $q_T^\circ \prec q_T'^\circ$ holds true for the rule occurrences q'_T and q_T in $\langle\!\langle \mathfrak{X} \fatsemi \mathfrak{Y} \rangle\!\rangle$.

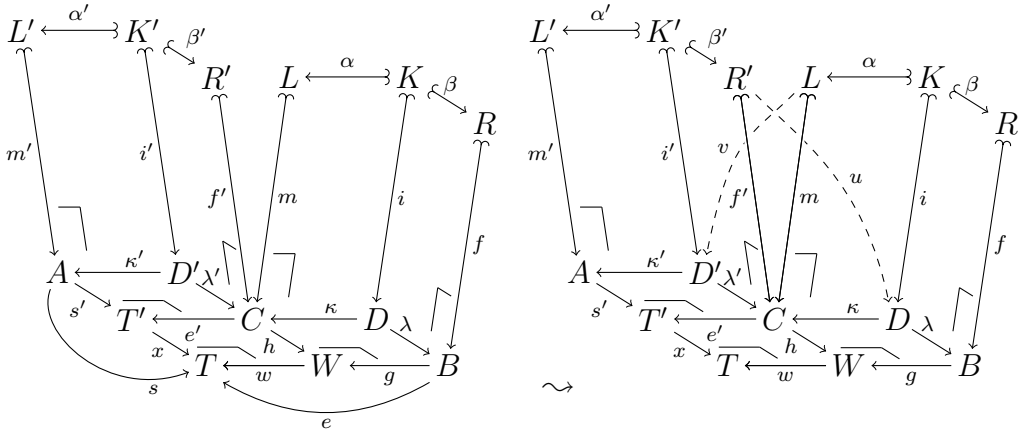


Figure 108: Sequential independence in a trace process of length two

Proof idea. Consider the pair of sequential-independent direct derivation diagrams and its trace process in the left hand diagram of Figure 108. This means that $q'_T = (s \circ m') \leftarrow\!\alpha'\!\rightarrow (s \circ m' \circ \alpha') \leftarrow\!\beta'\!\rightarrow (x \circ e' \circ f)$ is the rule occurrence of q' and $q_T = (x \circ e' \circ m) \leftarrow\!\alpha\!\rightarrow (x \circ e' \circ m \circ \alpha') \leftarrow\!\beta\!\rightarrow (e \circ f)$ is the one of the rule q. Now the desired follows from the fact that $\leftarrow\!i'\!\rightarrow\!\leftarrow\!\lambda'\!\rightarrow$ is an $\mathscr{M}$-final pullback complement of $\leftarrow\!\beta'\!\rightarrow\!\leftarrow\!f'\!\rightarrow$, and that $\leftarrow\!\kappa\!\rightarrow\!\leftarrow\!i\!\rightarrow$ is an $\mathscr{M}$-final pullback complement of $\leftarrow\!m\!\rightarrow\!\leftarrow\!\alpha\!\rightarrow$, yielding pullback squares of the form ${}^{K'}_{D'}\!\downarrow\!\overrightarrow{\sqsupseteq}\!\downarrow^{R'}_{C}$ and ${}^{L}_{C}\!\downarrow\!\overleftarrow{\sqsubseteq}\!\downarrow^{K}_{D}$, respectively.

First assume that the pair of direct derivation diagrams is sequential-independent as in the right hand diagram of Figure 108. Now, following the common practice of leaving the involved monomorphisms of subobjects implicit since all intersections are taken in $\mathrm{Sub}(T)$, the inclusion $R' \sqcap L \sqsubseteq K'$ holds because $\leftarrow\!i'\!\rightarrow\!\leftarrow\!\lambda'\!\rightarrow$ is an $\mathscr{M}$-final pullback complement of $\leftarrow\!\beta'\!\rightarrow\!\leftarrow\!f'\!\rightarrow$, and, using a completely analogue argument, also $L \sqcap R' \sqsubseteq K$ must hold; in other words both $q' \not\prec q$ and $q^\circ \not\prec q'^\circ$ hold true.

Conversely, assume that $q' \not\prec q$ and $q^\circ \not\prec q'^\circ$ hold true. Then the inclusion $R' \sqcap L \sqsubseteq K'$ holds, and since $\leftarrow\!i'\!\rightarrow\!\leftarrow\!\lambda'\!\rightarrow$ is an $\mathscr{M}$-final pullback complement of $\leftarrow\!\beta'\!\rightarrow\!\leftarrow\!f'\!\rightarrow$, also $L \sqsubseteq D'$ must be true, which yields a unique morphism $v\colon L \to D$ satisfying $m = \lambda' \circ v$. Using a completely analogous argument, there must exist a unique morphism $u\colon R' \to D$ satisfying $f' = \kappa \circ u$. □

After these two preliminary results, the proof of the pivotal proposition, namely Proposition 4.12, will establish the fact that processes provide an

alternative means to determine switching couples. The proposition says that a pair of sequential-independent derivations forms a switching couple (see Figure 109) if and only if their trace processes are isomorphic. The proof is as

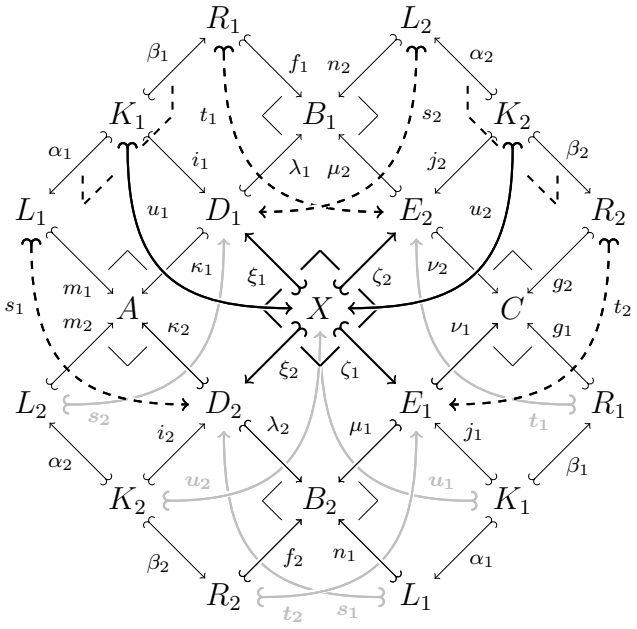


Figure 109: Filling the cell between canonical switchings

follows.

Proof sketch of Proposition 4.12. Given a switching couple as in Figure 109, use Lemma C.10 to conclude that the trace process of either one of the sequential-independent derivations is obtained by taking the pushout of A and C over X, i.e. by taking the pushout of the span $A \leftarrow \kappa_1 \circ \xi_1 - X - \zeta_2 \fatsemi \nu_2 \rightarrow C$, which is equal to $A \leftarrow \kappa_2 \circ \xi_2 - X - \zeta_1 \fatsemi \nu_1 \rightarrow C$. As pushouts are unique (up to isomorphism), the resulting trace processes are can be shown to be isomorphic.

The converse direction starts from the diagram in Figure 109, where w.l.o.g. the isomorphism between the type objects of the two trace processes is the identity id_T. In the diagram of this figure, identity arrows between two occurrences of the same object are omitted. The proof obligation amounts to constructing all arrows of the diagram in Figure 109 which are not present in Figure 110.

First construct X as a pullback $D_1 \leftarrow \xi_1 \twoheadrightarrow X \twoheadleftarrow \zeta_2 \rightarrow E_2$ of $D_1 \twoheadleftarrow \lambda_1 \rightarrow B_1 \leftarrow \mu_2 \twoheadrightarrow E_2$. The pullback yields $u_2 \colon K_2 \hookleftarrow X$ and $u_1 \colon K_1 \hookleftarrow X$, and the two squares $\begin{smallmatrix} L_2 & \leftarrow \\ D_1 & \downarrow \end{smallmatrix}$ $\begin{smallmatrix} \leftarrow & \\ \downarrow & K_2 \\ X & \end{smallmatrix}$

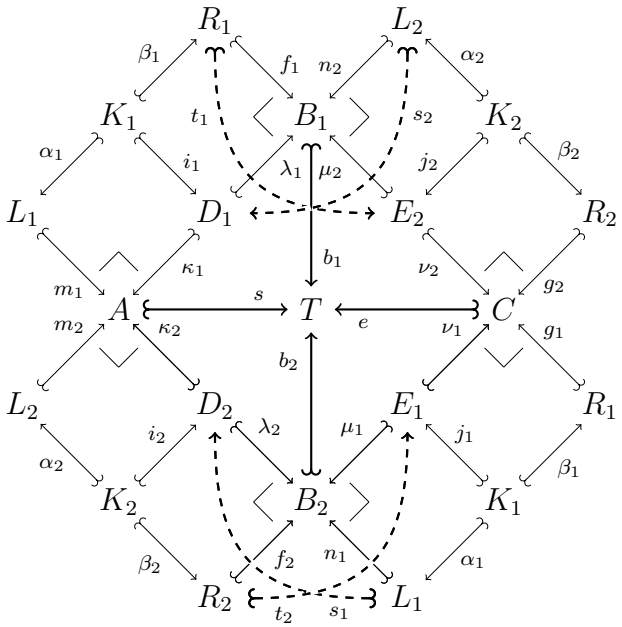


Figure 110: Sequential-independent derivations yielding the same trace process

and ${}^{K_1}_{X}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{R_1}_{E_2}$ are pushouts and hence pullbacks. Further there exists $\xi_2 \colon X \hookleftarrow D_2$ since $X \sqcap L_2 \sqsubseteq K_2$ where the involved subobjects are left implicit; similarly there is a unique "inclusion"-morphism $\zeta_1 \colon X \hookleftarrow E_1$. The square ${}^{D_1}_{A}\!\downarrow\!{}^{\leftarrow}_{\leftarrow}\!\downarrow{}^{X}_{D_2}$ is a pushout square by the Pushout Lemma since ${}^{A}_{L_2}\!\uparrow\!{}^{\leftarrow}_{\leftarrow}\!\uparrow{}^{D_2}_{K_2}$ and ${}^{D_1}_{L_2}\!\uparrow\!{}^{\leftarrow}_{\leftarrow}\!\uparrow{}^{X}_{K_2}$ are pushout squares; further this square is also a pullback. With an analogous argument, also ${}^{X}_{E_1}\!\downarrow\!{}^{\rightarrow}_{\rightarrow}\!\downarrow{}^{E_2}_{C}$ is a pushout and a pullback.

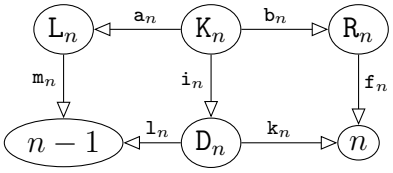
It remains to show that the square ${}^{D_2}_{B_2}\!\downarrow\!{}^{\leftarrow}_{\leftarrow}\!\downarrow{}^{X}_{E_1}$ is a (pushout and a) pullback. However, this is the case since both ${}^{D_2}_{K_2}\!\uparrow\!{}^{\rightarrow}_{\rightarrow}\!\uparrow{}^{B_2}_{R_2}$ and ${}^{X}_{K_2}\!\uparrow\!{}^{\rightarrow}_{\rightarrow}\!\uparrow{}^{E_1}_{R_2}$ are pushout squares. □

A direct consequence of Proposition 4.12 is that switch-equivalent derivations yield isomorphic processes. However Theorem 4.11 states also the converse, namely for a given pair $\mathfrak{X}, \mathfrak{Z} \colon A \Longrightarrow B$ of derivations, if the processes $\langle\![\mathfrak{X}]\!\rangle$ and $\langle\![\mathfrak{Y}]\!\rangle$ are isomorphic $\langle\![\mathfrak{X}]\!\rangle \cong \langle\![\mathfrak{Y}]\!\rangle$, then also $\mathfrak{X}$ and $\mathfrak{Z}$ are switch-equivalent – up to isomorphism. The intended notion of isomorphism of derivations is closely related to abstraction equivalence of [Baldan, 2000]. More precisely, two derivations $\mathfrak{X}_1, \mathfrak{X}_2 \colon A \Longrightarrow B$ are isomorphic in $\textsc{dpo}(\mathbb{C})$ if there is an isomorphism between the diagrams which is the identity on all rule components and the domain and codomain objects A and B. Here, as usual, derivation

diagrams are identified with the "obvious" diagrams in the category $\mathbb{C}$ as made more formal in the following definition.

☼ Definition C.12 (Isomorphism of derivations) Let $\mathfrak{X}\colon A \to B$ be a derivation. Its *shape* $\wr\mathfrak{X}\int$ is defined by recursion on the length of the derivation $\mathfrak{X}$ as follows:

The shape of the identity $\mathfrak{I}_A\colon A \to A$ in the derivation category $\mathrm{DPO}(\mathbb{C})$ is the one node graph $\wr\mathfrak{I}_A\int = \langle\{0\}, \varnothing, \varnothing, \varnothing\rangle$. The shape of a derivation $\mathfrak{Z}\mathcal{Z}\colon A \to B$ of length $n > 0$ where $\mathcal{Z}$ is a direct derivation diagram, is the union $\wr\mathfrak{Z}\mathcal{Z}\int := \wr\mathfrak{Z}\int \cup G_n$ where G_n is the illustrated graph.



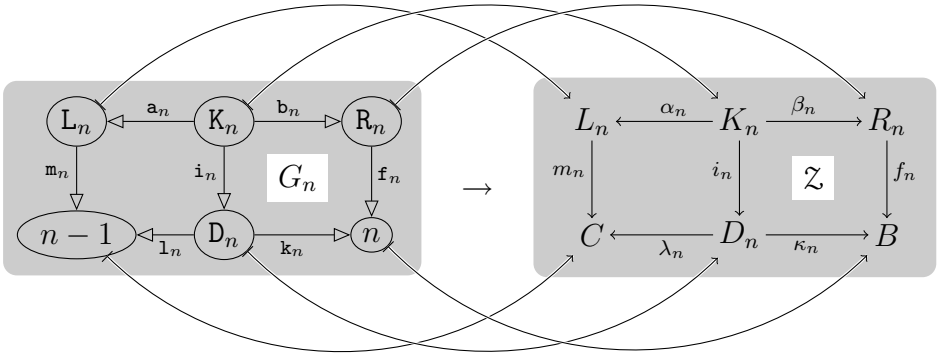
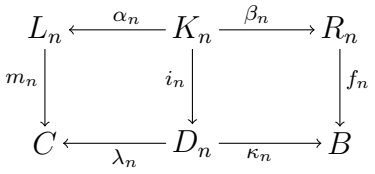


Figure 111: Formal diagram of a direct derivation diagram

The *diagram of a derivation* $\mathfrak{X}\colon \wr\mathfrak{X}\int \to \mathbb{C}$ is a diagram from the graph $\wr\mathfrak{X}\int$, i.e. an object of $\mathbb{Diag}[\wr\mathfrak{X}\int, \mathbb{C}]$ (see Example A.10). It is defined recursively on the length of $\mathfrak{X}$. The diagram of the identity $\mathfrak{I}_A\colon A \to A$ maps the single node of $\wr\mathfrak{I}_A\int$ to $A \in \mathbb{C}$. The diagram $\mathfrak{Z}\mathcal{Z}\colon \wr\mathfrak{Z}\mathcal{Z}\int \to \mathbb{C}$ of a derivation $\mathfrak{Z}\mathcal{Z}\colon A \to B$ of length n where $\mathcal{Z}$ is the illustrated direct derivation diagram, coincides with $\mathfrak{Z}\colon \wr\mathfrak{Z}\int \to \mathbb{C}$ on the subgraph $\wr\mathfrak{Z}\int \subseteq \wr\mathfrak{Z}\mathcal{Z}\int$ and on G_n it is defined as illustrated in Figure 111.



Two parallel derivations $\mathfrak{X}, \mathfrak{Z}\colon A \to B$ in $\mathrm{DPO}(\mathbb{C})$ of are *isomorphic*, written $\mathfrak{X} \equiv \mathfrak{Z}$, if they have the same length $n \in \mathbb{N}$ and (hence) the same shape $\wr\mathfrak{X}\int = \wr\mathfrak{Z}\int =: \langle V, E, \mathrm{src}, \mathrm{tgt}\rangle$, i.e. if their diagrams are objects of $\mathbb{Diag}[\wr\mathfrak{X}\int, \mathbb{C}]$, and additionally there is an isomorphism $\mathfrak{i}\colon \mathfrak{X} \to \mathfrak{Z}$ in $\mathbb{Diag}[\wr\mathfrak{X}\int, \mathbb{C}]$ such that for all nodes $v \in V \setminus (\{1, \ldots, n-1, \mathtt{D}_1, \ldots, \mathtt{D}_n\})$ $\mathfrak{i}_v$ is an identity. ☼

Now, the following lemma can be derived using the general results about occurrence grammars of Appendix D.

Lemma C.13 (Switch-equivalence of executions) Given a concatenable process $[Y] = [\langle s\colon A \to T, Q, e\colon B \to T\rangle]$, all executions are switch equivalent to each other, i.e. given two executions $\mathfrak{X}, \mathfrak{Z}\colon s \Rightarrow^{!Q} e$ of Y, then $\mathfrak{X} \equiv\approx_{\mathrm{sw}}\equiv \mathfrak{Z}$ holds where $\equiv\approx_{\mathrm{sw}}\equiv$ is relation composition of the relations $\equiv$ and $\approx_{\mathrm{sw}}$.

Proof idea. The proof is by induction on the number of transpositions in the sequence of the applied rules in $\mathfrak{X}$ and $\mathfrak{Z}$. The base case is the one in which $\mathfrak{X}$ and $\mathfrak{Z}$ apply the same sequence of rules. Then it is straightforward to verify that $\mathfrak{X} \equiv \mathfrak{Z}$.

For the induction step, given two derivations $\mathfrak{X}$ and $\mathfrak{Z}$, these can be decomposed as $\mathfrak{X} \equiv \mathfrak{X}_1 \fatsemi \mathcal{V}_1\mathcal{W}_2 \fatsemi \mathfrak{X}_2$ and $\mathfrak{Z} \equiv \mathfrak{Z}_1 \fatsemi \mathcal{W}_1\mathcal{V}_2 \fatsemi \mathfrak{Z}_2$ such that $\mathfrak{X}_1 \equiv \mathfrak{Z}_1$ and $\mathfrak{X}_2 \equiv\approx_{\mathrm{sw}}\equiv \mathfrak{Z}_2$ by the induction hypothesis; further, w.l.o.g. $\mathcal{W}_1\mathcal{V}_2$ and $\mathcal{V}_1\mathcal{W}_2$ are parallel in $\mathrm{DPO}(\mathbb{C}\downarrow T)$, i.e. $\mathcal{W}_1\mathcal{V}_2, \mathcal{V}_1\mathcal{W}_2\colon c_1 \Rightarrow\!\!\!\!\Rightarrow c_2$.

Using the results of Appendix D, all the linearizations of the rule set Q which are induced by these derivations are compatible with (co-)asymmetric conflict. Hence Scholium C.9 allows to decompose $[Y]$ as $[Y] = [Y_1] \fatsemi [\bar{Y}] \fatsemi [Y_2]$ such that $[\bar{Y}] = [\langle\!|\lrcorner|\circ(\mathcal{V}_1\mathcal{W}_2)|\!\rangle] = [\langle\!|\lrcorner|\circ(\mathcal{W}_1\mathcal{V}_2)|\!\rangle]$ where the forgetful functor $|\lrcorner|\colon \mathbb{C}\downarrow T \to \mathbb{C}$ gives the underlying $\mathbb{C}$-derivations. Now apply Lemma C.10 to conclude that $\mathcal{V}_1\mathcal{W}_2$ and $\mathcal{W}_1\mathcal{V}_2$ are two pairs of sequential-independent direct derivations; further Proposition 4.12 yields that they actually form a switching couple because the equation $[\langle\!|\lrcorner|\circ(\mathcal{V}_1\mathcal{W}_2)|\!\rangle] = [\langle\!|\lrcorner|\circ(\mathcal{W}_1\mathcal{V}_2)|\!\rangle]$ holds. This however implies that $\mathfrak{X}$ and $\mathfrak{Z}$ are switch-equivalent up to isomorphism, i.e. the desired $\mathfrak{X} \equiv\approx_{\mathrm{sw}}\equiv \mathfrak{Z}$ is established. □

Finally, this lemma can be used to complete the proof of Theorem 4.14. As a direct consequence, there is an isomorphism between $\mathrm{CP}(\mathbb{C})$ and $\mathrm{DPO}(\mathbb{C})^{\sim}_{\equiv}$ where the latter category arises from the trace category $\mathrm{DPO}(\mathbb{C})^{\sim}$ by identifying pairs of isomorphic derivations (see also [Baldan et al., 2006a, Theorem 25]).

Facts about occurrence grammars

D

This section gives a rather complete account of the the results about occurrence grammars. For a natural number $n \in \mathbb{N}$, define $\{\!|n|\!\} = \{0, \ldots, n-1\}$, which for the case of $n = 0$ means $\{\!|0|\!\} = \varnothing$. Further, fix some weakly $\mathscr{M}\omega$-adhesive category $\mathbb{C}$ to which all objects and morphisms belong unless stated otherwise. Moreover, $\mathbb{C}$ is assumed to have final $\mathscr{M}$-pullback complements of pairs of composable $\mathscr{M}$-morphisms into objects with at most countably many different subobjects.

LEMMA D.1 (Causal acyclicity) Let $O = \langle Q, s\colon S \hookleftarrow T \rangle$ be an occurrence grammar. Then the causal relation $\prec$ is acyclic.

Proof. Let $q \in Q$ be a rule name. Now the proof proceeds by induction on $n \in \mathbb{N}_0$ and shows that there is no cycle of length n.

$n = 0$ In this case it is enough to show that $\prec$ is irreflexive. Suppose for contradiction that $q \prec q$. Then by definition $q \in \lfloor q \rfloor$ and $q \nearrow q$ hold, from which acyclicity of $\nearrow|_{\lfloor q \rfloor}$ follows, which contradicts the assumption that O is an occurrence grammar.

$n > 0$ Suppose for contradiction that there is a a (nontrivial) cycle in causality of length n, such that say $q = q_0 \prec q_1 \preceq q_n = q$. This however means that $q \preceq q_1$ and $q_1 \preceq q$, and hence antisymmetry of $\preceq$ implies $q = q_1$ whence $q = q_0 \prec q_1 = q$, which cannot be, as shown in the base case.

□

COROLLARY D.2 For each rule q of an occurrence grammar $\langle Q, s\colon S \hookleftarrow T \rangle$, the equation $r_q \sqcap l_q = k_q$ holds in $\mathrm{Sub}(T)$.

Proof. Clearly $k_q \sqsubseteq r_q \sqcap l_q$; further $q \not\prec q$, i.e. $r_q \sqcap l_q \sqsubseteq k_q$. □

COROLLARY D.3 (Unique causes) Let $O = \langle Q, s\colon S \hookleftarrow T \rangle$ be an occurrence grammar, and let $q, q' \in Q$ such that $q \neq q'$. Then $r_q \sqcap r_{q'} \sqsubseteq k_q$ or $r_q \sqcap r_{q'} \sqsubseteq k_{q'}$.

Proof. Because of Lemma D.1, not both of $q \prec q'$ and $q' \prec q$ can hold. Hence w.l.o.g. assume that $q \not\prec q'$, i.e. $r_q \sqcap l_{q'} \sqsubseteq k_q$. Further, since O is an

occurrence grammar, the inclusion $r_q \sqcap r_{q'} \sqsubseteq k_q \sqcup k_{q'}$ holds, i.e. it is the case that $r_q \sqcap r_{q'} = (r_q \sqcap r_{q'}) \sqcap (k_q \sqcup k_{q'})$. Now derive

$$\begin{aligned}
r_q \sqcap r_{q'} &= (r_q \sqcap r_{q'}) \sqcap (k_q \sqcup k_{q'}) && \text{(the last equation)} \\
&= (r_q \sqcap r_{q'} \sqcap k_q) \sqcup (r_q \sqcap r_{q'} \sqcap k_{q'}) && \text{(distributivity)} \\
&= (r_{q'} \sqcap k_q) \sqcup (r_q \sqcap k_{q'}) && (k_q \sqsubseteq r_q \text{ and } k_{q'} \sqsubseteq r_{q'}) \\
&\sqsubseteq (r_{q'} \sqcap k_q) \sqcup (r_q \sqcap l_{q'}) && (k_{q'} \sqsubseteq l_{q'}) \\
&\sqsubseteq (r_{q'} \sqcap k_q) \sqcup k_q && (q \not\prec q') \\
&= k_q.
\end{aligned}$$

Note that whenever distributivity is used, the relevant joins are $\mathscr{M}$-subobjects and distributivity holds by Proposition 7.18. □

A first step towards a static characterization of derivable objects is the following lemma, which extends causal completeness from left hand sides of rules to subobjects with finite causes, and in particular concurrent subobjects.

LEMMA D.4 (Causal completeness) Given an occurrence grammar $O = \langle Q, s\colon S \hookrightarrow T\rangle$ and an $\mathscr{M}$-subobject $[a] \in \mathrm{Sub}_{\mathscr{M}}(T)$ with a finite number of causes $\lfloor a \rfloor$, then the inclusion $a \sqsubseteq s \sqcup \bigsqcup_{q\in\lfloor a\rfloor} r_q$ holds.

Proof. By type minimality, $a = a \sqcap \left(s \sqcup \bigsqcup_{q\in Q} r_q\right)$; further, using distributivity,

$$a = (a \sqcap s) \sqcup \left(\bigsqcup_{q\in Q} a \sqcap r_q\right). \tag{18}$$

Now, let $q \in Q$ be any rule. If $q \in \lfloor a \rfloor$, then clearly $a \sqcap r_q \sqsubseteq s \sqcup \bigsqcup_{q\in\lfloor a\rfloor} r_q$; so assume that $q \notin \lfloor a \rfloor$, and in particular $r_q \sqcap a \sqsubseteq k_q$ whence $r_q \sqcap a \sqsubseteq a \sqcap k_q \sqsubseteq a \sqcap l_q$.

In the latter, second case, it is enough to show that $a \sqcap l_q \sqsubseteq s \bigsqcup_{q\in\lfloor a\rfloor} r_q$; for this, proceed by induction on the depth of the rule q, i.e. on the maximal length of all chains $q_1 \prec \cdots \prec q_n = q$ of rules "before" q.

$n = 0$: This means that $\lfloor l_q \rfloor = \varnothing$; hence $a \sqcap l_q \sqsubseteq l_q \sqsubseteq s \sqsubseteq s \bigsqcup_{q\in\lfloor a\rfloor} r_q$.

$n \rightsquigarrow n+1$: The inclusion $l_q \sqsubseteq s \sqcup \bigsqcup_{q'\in\lfloor l_q\rfloor} r_{q'}$ holds by causal completeness; hence also

$$a \sqcap l_q \sqsubseteq (a \sqcap s) \sqcup \bigsqcup_{q'\in\lfloor l_q\rfloor} a \sqcap r_{q'}$$

must hold. The induction hypothesis applies to all rules $q' \in \lfloor l_q \rfloor$, and hence not only $(a \sqcap s) \sqsubseteq s \sqcup \bigsqcup_{q\in\lfloor a\rfloor} r_q$ but all $a \sqcap r_{q'} \sqsubseteq s \sqcup \bigsqcup_{q\in\lfloor a\rfloor} r_q$. Using the properties of joins, $a \sqcap l_q \sqsubseteq s \sqcup \bigsqcup_{q\in\lfloor a\rfloor} r_q$.

□

LEMMA D.5 (Finite configurations) Let $O = \langle Q, s\colon S \hookrightarrow T\rangle$ be an occurrence grammar, let $Q' \subseteq Q$ be a finite set, which is downward closed w.r.t. causality, i.e. such that $\lfloor Q' \rfloor = Q'$.

Then Q' is a configuration if and only if there is a linearization of Q' that is compatible with $\nearrow$, i.e. if and only if there is a sequence $\langle q_i \rangle_{i \in \{\!|Q'|\!\}}$ of rules, such that the equation $Q' = \{q_i \mid i \in \{\!|Q'|\!\}\}$ holds and $q_i \nearrow q_j$ implies $i < j$ for all $i, j \in \{\!|Q'|\!\}$.

Proof. The proof is a routine induction on the length of Q'. □

LEMMA D.6 Let $O = \langle Q, s\colon S \hookrightarrow T\rangle$ be an occurrence grammar, let $n \geq 0$ be a natural number, let $Q' \subseteq Q$ be a set of rules with $|Q'| = n$, and let $(A \xrightarrow{a} T)$ be derived using each rule in Q' exactly once, written $(S \xrightarrow{s} T) \Rightarrow^{!Q'} (A \xrightarrow{a} T)$. Then the following are true.

1. The morphism a belongs to $\mathscr{M}$.
2. The set Q' is a configuration.
3. The subobject a is covered by the right hand sides of Q', i.e. inclusion $a \sqsubseteq s \sqcup \bigsqcup_{q \in Q'} r_q$ holds in $\mathrm{Sub}(T)$.
4. All causes of a are within Q', i.e. $\lfloor a \rfloor \subseteq Q'$.
5. All co-causes of a are outside of Q', i.e. the equation $\ulcorner a \urcorner \cap Q' = \varnothing$ holds.
6. The subobject a is the greatest $\mathscr{M}$-subobject with the previous three properties, i.e. any $\mathscr{M}$-subobject $[x] \in \mathrm{Sub}_{\mathscr{M}}(T)$ satisfying

$$x \sqsubseteq s \sqcup \bigsqcup_{q \in Q'} r_q, \qquad \lfloor x \rfloor \sqsubseteq Q', \text{ and } \qquad \ulcorner x \urcorner \cap Q' = \varnothing$$

   is covered by a, i.e. $a \sqsupseteq x$.

Proof. The base case $Q' = \varnothing$ is rather trivial.

1. By definition s is an $\mathscr{M}$-morphism.
2. The empty set $\varnothing$ is a finite configuration.
3. Also $s \sqsubseteq s \sqcup \bigsqcup_{q \in Q'} r_q = s$.

4. The equation $\lfloor s \rfloor = \varnothing$ holds by definition.

5. As another triviality, $\ulcorner s \urcorner \cap \varnothing = \varnothing$.

6. Given any subobject $[x] \in \mathrm{Sub}(T)$, the required $x \sqsubseteq s$ already follows from $x \sqsubseteq s \sqcup \bigsqcup_{q \in \varnothing} r_q$.

Now, for the induction step, suppose there is a derivation

$$(S \xrightarrow{s} T) \models_{\cdot}^{!Q''} \Rightarrow (A \xrightarrow{a} T) \models_{\cdot}^{q} \Rightarrow (B \xrightarrow{b} T)$$

such that $q = (l_q \leftarrow \alpha_q \rightarrow k_q \leftarrow \beta_q \rightarrow r_q) \notin Q''$ and the last derivation step is witnessed by the left one of the diagrams in Figure 112. It is enough to show all the six

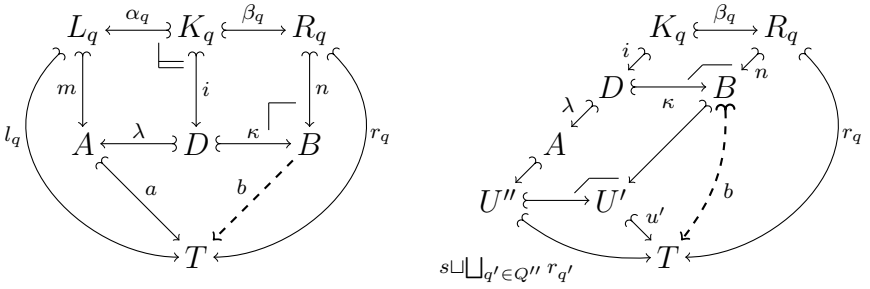


Figure 112: Main constructions for the proof of Lemma

items for $Q' := Q'' \uplus \{q\}$.

1. First, the inclusion $k_q \sqsubseteq r_q \sqcap a$ follows from the two inclusions $k_q \sqsubseteq r_q$ and $k_q \sqsubseteq l_q \sqsubseteq a$. Further, $r_q \sqcap a \sqsubseteq k_q$ follows from the induction hypothesis $q \notin \lfloor a \rfloor$, and hence actually $k_q = r_q \sqcap a$ holds. Moreover $a \sqsubseteq s \sqcup \bigsqcup_{q' \in Q''} r_{q'} =: u''$ is true by the induction hypothesis, and hence the union $r_q \sqcup (s \sqcup \bigsqcup_{q' \in Q''} r_{q'}) =: u'$ arises by pushout over K_q as indicated in the right hand diagram in Figure 112.

   Now, the join $u' = r_q \sqcup (s \sqcup \bigsqcup_{q' \in Q''} r_{q'})$ is actually an $\mathscr{M}$-subobject (which is a consequence of either dissectability of the occurrence grammar O or $\mathscr{M}$-effectivity of unions in $\mathbb{C}$). Hence also b must be an $\mathscr{M}$-morphism, which follows by pushout splitting as sketched in Figure 112 and the fact that $\mathscr{M}$-morphisms are closed under composition.

2. First, verify that Q' is downward closed. For this it is enough to show that $\lfloor q \rfloor \setminus \{q\} \subseteq Q''$ since Q'' is downward closed by the induction hypothesis. However for this it already suffices to show that $\llcorner q \lrcorner \subseteq Q''$.

Now take an arbitrary direct cause $q'' \in \llcorner l_q \lrcorner = \llcorner q \lrcorner$; then already $q'' \in \llcorner a \lrcorner$ holds, since $l_q \sqsubseteq a$. The induction hypothesis $\llcorner a \lrcorner \subseteq Q''$ implies that $q'' \in Q''$; this however already means that that Q' is downward closed.

In a second step, use Lemma D.5 to show that Q' is actually a configuration. For this, let $q'' \in \ulcorner l_q \urcorner$, i.e. $l_q \sqcap l_{q''} \not\sqsubseteq k_{q''}$. But then also $a \sqcap l_{q''} \not\sqsubseteq k_{q''}$ must hold, i.e. $q'' \in \ulcorner a \urcorner$ whence $q'' \notin Q''$ follows. This shows that $\ulcorner l_q \urcorner \cap Q'' = \varnothing$. However, $q \nearrow q$ does not hold either since $\nearrow|_{\lfloor q \rfloor}$ is irreflexive by definition. Because Q' is downward closed it also follows that there is no $q' \in Q'$ such that $q \preceq q'$.

Taking everything together, q can be appended to any linearization of Q'' which is compatible with $\nearrow$ and the result is a linearization of Q'. Lemma D.5 now implies that Q' is a finite configuration.

3. The inclusions $d \sqsubseteq a \sqsubseteq s \sqcup \bigsqcup_{q' \in Q''} r_{q'}$ have already been established and $b = d \sqcup r_q$ where both d and r_q are $\mathscr{M}$-morphisms; hence $b \sqsubseteq s \sqcup \bigsqcup_{q' \in Q'} r_{q'}$ (see also Figure 112).

4. As Q' is downward-closed, i.e. since $\lfloor Q' \rfloor = Q'$ holds, it is enough to show that $\llcorner b \lrcorner \subseteq Q'$. If q is not producing, i.e. if β_q is invertible, then $b = d$, and hence $\llcorner b \lrcorner \subseteq \llcorner d \lrcorner \subseteq \llcorner a \lrcorner \subseteq Q'' \subseteq Q'$.

   Otherwise, i.e. if q is producing, the equation $\llcorner r_q \lrcorner = \llcorner k_q \lrcorner \cup \{q\}$ hold, which can be justified as follows. First, the inclusion $\llcorner r_q \lrcorner \supseteq \llcorner k_q \lrcorner \cup \{q\}$ is trivial; second, to show $\llcorner r_q \lrcorner \subseteq \llcorner k_q \lrcorner \cup \{q\}$, let $q' \in \llcorner r_q \lrcorner \setminus \{q\}$, i.e. $q \neq q'$ and $r_{q''} \sqcap r_q \not\sqsubseteq k_{q''}$. Then use Corollary D.3 and conclude that $r_{q''} \sqcap r_q \sqsubseteq k_q$, i.e.

$$r_{q''} \sqcap k_q = r_{q''} \sqcap r_q \sqcap k_q = r_{q''} \sqcap r_q \not\sqsubseteq k_{q''}$$

   whence $q'' \in \llcorner k_q \lrcorner$.

   Using this observation and the two facts $d \sqsubseteq a$ and $b = d \sqcup r_q$, derive the following.

$$\begin{aligned} \llcorner b \lrcorner &= \llcorner d \lrcorner \cup \llcorner r_q \lrcorner \\ &= \llcorner d \lrcorner \cup \llcorner k_q \lrcorner \cup \{q\} \\ &= \llcorner d \lrcorner \cup \{q\} \\ &\subseteq Q' \end{aligned}$$

5. Similarly $\ulcorner b \urcorner = \ulcorner d \urcorner \cup \ulcorner r_q \urcorner$. Clearly $q \notin \ulcorner d \urcorner$; the induction hypothesis implies that $\ulcorner d \urcorner \cap Q'' = \varnothing$ and hence $\ulcorner d \urcorner \cap Q' = \varnothing$. Therefore it remains to show that $\ulcorner r_q \urcorner \cap Q' = \varnothing$.

Let $q' \in \ulcorner r_q \urcorner$. Now, $q \notin \ulcorner r_q \urcorner$ is a consequence of Corollary D.2, and hence $q' \neq q$. Now if $q \prec q'$ then $q' \notin Q''$ follows from the fact that Q'' is downward closed; hence the case that $q \not\prec q'$ is left, i.e. $l_{q'} \sqcap r_q \sqsubseteq k_q$ holds or equivalently, $l_{q'} \sqcap r_q = l_{q'} \sqcap r_q \sqcap k_q = l_{q'} \sqcap k_q$. From this it follows that q' is actually a co-cause of k_q, in other words $q' \in \ulcorner k_q \urcorner \subseteq \ulcorner d \urcorner$. As $\ulcorner d \urcorner \cap Q'' = \varnothing$ has already been shown above, the desired $q' \notin Q'$ follows since $q' \neq q$.

6. Let $[x] \in \mathrm{Sub}_{\mathscr{M}}(T)$ be an $\mathscr{M}$-subobject which satisfies all three of $x \sqsubseteq s \sqcup \bigsqcup_{q \in Q'} r_q$, $\lfloor x \rfloor \subseteq Q'$, and $\ulcorner x \urcorner \cap Q' = \varnothing$ hold. Now the proof obligation is to show that $[x]$ is contained in $[b]$, i.e. that $x \sqsubseteq b$ holds.

   First, since $x \sqsubseteq s \sqcup \bigsqcup_{q' \in Q'} r_{q'}$ the equality $x \sqcap (s \sqcup \bigsqcup_{q' \in Q'} r_{q'}) = x$ holds.

$$\begin{aligned} x &= x \sqcap (s \sqcup \bigsqcup_{q' \in Q'} r_{q'}) \\ &= (x \sqcap s) \sqcup \bigsqcup_{q' \in Q'} (x \sqcap r_{q'}) \\ &= (x \sqcap r_q) \sqcup \underbrace{(x \sqcap s) \sqcup \bigsqcup_{q' \in Q''} (x \sqcap r_{q'})}_{y} \end{aligned} \tag{19}$$

   Clearly $\llcorner y \lrcorner \subseteq \llcorner x \lrcorner \subseteq Q'$ holds, and next also $q \notin \llcorner y \lrcorner$ will be derived. For this, recall that q can be appended to any linearization of Q'' (as shown above); hence $q \not\prec q'$ holds for all $q' \in Q''$.

$$\begin{aligned} r_q \sqcap y &= r_q \sqcap \left((x \sqcap s) \sqcup \bigsqcup_{q' \in Q''} (x \sqcap r_{q'}) \right) \\ &= (x \sqcap s \sqcap r_q) \sqcup \bigsqcup_{q' \in Q''} (x \sqcap r_{q'} \sqcap r_q) \\ &\sqsubseteq (x \sqcap k_q) \sqcup \bigsqcup_{q' \in Q''} (x \sqcap k_q) \qquad (q \not\prec q' \text{ and } \llcorner s \lrcorner = \varnothing) \\ &\sqsubseteq k_q \end{aligned}$$

   Hence the inclusion $\llcorner y \lrcorner \subseteq Q''$ holds, and since Q'' is cause-closed, also $\lfloor y \rfloor \subseteq Q''$; further the assumptions on x imply the equation $\ulcorner y \urcorner \cap Q'' = \varnothing$ because both $\ulcorner y \urcorner \subseteq \ulcorner x \urcorner$ and $\ulcorner x \urcorner \cap Q'' = \varnothing$ hold. Using the induction hypothesis on a, the inclusion $y \sqsubseteq a$ is obtained. Further $q \notin \ulcorner y \urcorner$ holds,

i.e. $l_q \sqcap y \sqsubseteq k_q$ is true, whence $y \sqsubseteq d$ follows, which in turn implies $y = y \sqcap d$.

Continuing from Equation (19) and using that $d \sqsubseteq s \sqcup \bigsqcup_{q' \in Q''} r_{q'}$ derive the following equations.

$$
\begin{aligned}
x &= (x \sqcap r_q) \sqcup y && \text{(Equation (19))} \\
&= (x \sqcap r_q) \sqcup (d \sqcap y) \\
&= (x \sqcap r_q) \sqcup d \sqcap \left((x \sqcap s) \sqcup \bigsqcup_{q' \in Q''} (x \sqcap r_{q'}) \right) && \text{(Definition of } y\text{)} \\
&= (x \sqcap r_q) \sqcup \left(x \sqcap d \sqcap \left(s \sqcup \bigsqcup_{q' \in Q''} r_{q'} \right) \right) \\
&= (x \sqcap r_q) \sqcup (x \sqcap d) \\
&= x \sqcap (r_q \sqcup d) \\
&= x \sqcap b
\end{aligned}
$$

This however means that $x \sqsubseteq b$.

□

The most relevant corollary of this lemma is the safety of occurrence grammars.

Corollary D.7 (Safety of occurrence grammars) Given a *consuming* occurrence grammar $O = \langle Q, s \colon S \hookleftarrow T \rangle$, each derived object $s \models\!\!\!\Rightarrow_Q a$ in O is an $\mathscr{M}$-morphism.

Proof. In a consuming occurrence grammar, $q \in \ulcorner l_q \urcorner$ holds for all $q \in Q$. Hence, let $Q' \subseteq Q$ a subset of rules, and $s \models\!\!\!\Rightarrow_{!Q'} a$ any derivation without repetition of rules. Now, $q' \notin \ulcorner a \urcorner$ holds for all $q' \in Q'$; hence q' cannot be applied again since $l_{q'} \sqsubseteq a$ would lead to the contradiction $q' \in \ulcorner a \urcorner$. □

Proposition D.8 (Characterization of derived objects) Given an occurrence grammar $O = \langle Q, s \colon S \hookleftarrow T \rangle$ and a finite configuration $C \subseteq Q$, there is a subobject which is derivable using each rule of C exactly once; i.e. there is some $[a] \in \mathrm{Sub}_{\mathscr{M}}(T)$ such that $[s] \models\!\!\!\Rightarrow_{!C} [a]$. Further the reached subobject $[a]$ can be described as the following join.

$$
a = \bigsqcup \left\{ [x] \in \mathrm{Sub}_{\mathscr{M}}(T) \mid \left(\lfloor x \rfloor \subseteq C \right) \;\&\; \left(\ulcorner x \urcorner \cap C = \varnothing \right) \right\}
$$

Proof. The proof is by induction on the length of C. For $|C| = 0$ this is trivial. Hence let $|C| > 0$ be a non-empty configuration, let $\langle q_i \rangle_{i \in \natural |C| \natural}$ be a linearization of C. Now, put $q := q_{|C|-1}$ and $C' := C \setminus \{q\}$. Then, the induction hypothesis implies that there is an object b which is derivable from s using each rule in C' exactly once, i.e. $s \models\!\!\!\!\!\!\cdot^{!C'}\!\!\Rightarrow b$, such that moreover both $\lfloor b \rfloor \subseteq C'$ and $\lceil b \rceil \cap C' = \varnothing$ hold. Further, since $l_q \sqsubseteq (s \sqcup \bigsqcup_{q' \in \lfloor q \rfloor} r_{q'}) \sqsubseteq (s \sqcup \bigsqcup_{q \in C'} r_q)$ holds, Lemma D.6.6 allows to establish the inclusion $l_q \sqsubseteq b$. Finally, as $\mathbb{C}$ has (enough) $\mathscr{M}$-final pullback complements of pairs of composable $\mathscr{M}$-morphisms and also pushouts of spans of $\mathscr{M}$-morphisms, a suitable object b is derivable.

Finally, the characterization of reachable objects as the greatest concurrent object with causes inside C and co-causes outside of C is a consequence of Lemma D.6 and Lemma D.4. □

The consequence of this characterization of reachable object is the coincidence of coverable and concurrent objects.

Corollary D.9 (Coverability of concurrent subobjects) Given an occurrence grammar $O = \langle Q, s\colon S \hookleftarrow T \rangle$ and a concurrent subobject $[a] \in \mathrm{Sub}_{\mathscr{M}}(T)$, there is some subobject $[b] \in \mathrm{Sub}_{\mathscr{M}}(T)$ such that $s \models\!\!\!\!\!\!\cdot^{!\lfloor a \rfloor}\!\!\Rightarrow b$ and $a \sqsubseteq b$.

Proof. Clearly, if $[a]$ is concurrent then $\lfloor a \rfloor$ is a finite configuration. Now apply Proposition D.8 to obtain a reachable object $s \models\!\!\!\!\!\!\cdot^{!\lfloor a \rfloor}\!\!\Rightarrow b$ such that $[b]$ is the maximal subobject with causes in $\lfloor a \rfloor$ and no co-causes in $\lfloor a \rfloor$. As a consequence the inclusion $a \sqsubseteq b$ must hold. □

The final fact about occurrence grammars that will be essential in the coreflection result is the Dissection Lemma.

Lemma D.10 (Dissection) Let $O = \langle Q, s\colon S \hookleftarrow T \rangle$ be an occurrence grammar in a weakly $\mathscr{M}\omega$-adhesive category. Then there is an enumeration $\{q_i \mid i \in \mathbb{N}\} = Q$ of all rules in Q such that id_T arises as the colimit of the chain

$$\left\{ s \sqcup \bigsqcup_{i<n} r_{q_i} \sqsubseteq s \sqcup \bigsqcup_{i \leq n} r_{q_i} \right\}_{n \in \mathbb{N}}.$$

Proof. First, as there is some enumeration $\{q'_j \mid j \in \mathbb{N}\}$ of the countable set Q, this enumeration can be assumed to be cause-closed as $\lfloor q \rfloor$ is finite for all $q \in Q$. Hence, by inserting $\lfloor q'_j \rfloor$ before each q'_j in some order that is compatible with causality, one obtains an enumeration $\{q_i \mid i \in \mathbb{N}\}$ of Q such that $\{q_i \mid i \leq n\} = \lfloor \{q_i \mid i \leq n\} \rfloor$ for all $n \in \mathbb{N}$.

Now (applying dissectability) each join $s \sqcup \bigsqcup_{i<n} r_{q_i}$ is an $\mathscr{M}$-subobject. Hence, the join of

$$\left\{ s \sqcup \bigsqcup_{i<n} r_{q_i} \sqsubseteq s \sqcup \bigsqcup_{i\leq n} r_{q_i} \right\}_{n\in\mathbb{N}}$$

can be computed via a colimit of a chain

$$\left\{ S \sqcup \bigsqcup_{i<n} R_{q_i} \hookleftarrow{m_n}\hookrightarrow S \sqcup \bigsqcup_{i\leq n} R_{q_i} \right\}_{n\in\mathbb{N}}$$

of $\mathscr{M}$-morphisms in $\mathbb{C}$ where the expression $S \sqcup \bigsqcup_{i<n} R_{q_i}$ is used to denote the domain of some (chosen) $\mathscr{M}$-morphism u_n such that $[u_n] = [s] \sqcup \bigsqcup_{i<n} [r_{q_i}]$, and $m_n \colon u_n \hookrightarrow u_{n+1}$ in $\mathbb{C}{\downarrow}T$ is the witness of the above inclusions. The properties of joins imply that the resulting join is equal to $s \sqcup \bigsqcup_{q\in Q} r_q = \mathrm{id}_T$. □

Establishing the unfolding as a coreflection

E This section gives a formal presentation of the theorem which establishes the category of occurrence grammars as a coreflective subcategory of the category of typed grammars. The two main properties of the unfolding construction is that it produces occurrence grammars which are based on fair unfolding sequences and that they are irredundant w.r.t. the folding morphism (see Lemma E.9). After these and other facts that are consequences of the proper construction of unfoldings and results about occurrence grammars, Section E.1 presents the main results concerning retyping functors and grammar morphisms. Finally Section E.2 concludes with the proof of the Coreflection Theorem.

☼ Definition E.1 (Unfolding sequences) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $T \in \mathbb{C}$, and let $G = \langle Q, s\colon S \to T\rangle$ be an $\mathscr{M}$-linear T-typed grammar. Then for each $n \in \mathbb{N}_0$ a *finite unfolding sequences for G of length n* is inductively defined as a triple of sequences $\big\langle \{U_i\}_{i\leq n}, \{\lambda_i\}_{i\leq n}, \{t_i\}_{i<n}\big\rangle$ where each U_i is an $\mathscr{M}$-linear T_i-typed grammar $U_i = \langle Q_i, s_i\colon S \hookleftarrow T_i\rangle$, called a *partial unfolding*, each λ_i is a $\mathbb{C}$-morphism $\lambda_i\colon T_i \to T$, referred to as the *folding morphism of U_i*, and each $t_i\colon T_i \hookleftarrow T_{i+1}$ is the *inclusion morphism* which embeds U_i into U_{i+1}.

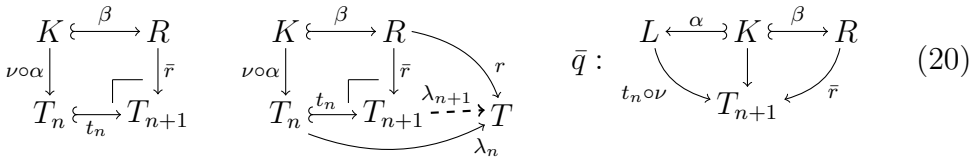
BASE CASE For $n = 0$, the grammar U_0 is defined as $U_0 := \langle \varnothing, \mathrm{id}_S\colon S \to S\rangle$, and the morphism λ_0 is defined as $\lambda_0 := s\colon S \to T$.

INDUCTIVE CASE Let $\big\langle \{U_i\}_{i\leq n}, \{\lambda_i\}_{i\leq n}, \{t_i\}_{i<n}\big\rangle$ be a finite unfolding sequence for G of length n. By definition, any extension of theses sequences to an unfolding sequence for G of length $n+1$ is obtained by means of the following procedure.

Choose any $\mathscr{M}$-SPO derivation $s_n \mathrel{\rightleftharpoons}_{Q_n}\!\Rightarrow a$ in the grammar U_n, any rule $q = (l \leftarrow\!\alpha\!- k -\!\beta\!\rightarrow r) \in Q$ of the grammar G, and any $\mathscr{M}$-morphism $m\colon l \hookrightarrow (\lambda_n \circ a)$ such that there is no rule $q' = (l' \leftarrow k' \rightarrow r') \in Q_n$ which satisfies both $\lambda_n \circ l' = l$ and $\Sigma_{\lambda_n}(q') = q$.

Now, the (left hand side of the) new rule occurrence is $\nu := a \circ m$. To obtain the new inclusion morphism $t_n\colon T_n \hookleftarrow T_{n+1}$, take a $\mathbb{C}$-pushout $T_n \hookleftarrow\!{\scriptstyle t_n}\!\rightarrow T_{n+1} \leftarrow\!{\scriptstyle \bar{r}}\!- R$ of the $\mathbb{C}$-span $T_n \leftarrow\!{\scriptstyle \nu\circ\alpha}\!- K \hookleftarrow\!{\scriptstyle \beta}\!\rightarrow R$ where $K = |k|_T$ and $R = |r|_T$ are the domains of k and r, respectively; this yields a pushout square $\begin{smallmatrix} K & \overrightarrow{} \\ R & \overrightarrow{} \end{smallmatrix}\!\downarrow\!\!\downarrow\!\begin{smallmatrix} T_n \\ T_{n+1} \end{smallmatrix}$ in $\mathbb{C}$. Moreover the new folding morphism $\lambda_n\colon T_{n+1} \to T$ arises as the unique mediating morphism satisfying both $\lambda_n = \lambda_{n+1} \circ t_n$

and $r = \lambda_{n+1} \circ \bar{r}$.



(20)

The complete new rule occurrence is

$$\bar{q} := \Big((t_n \circ \nu) \leftarrow\!\alpha\!\rightarrow (t_n \circ \nu \circ \alpha) \leftarrow\!\beta\!\rightarrow \bar{r} \Big).$$

The new set of rules $Q_{n+1} := \{\bar{q}\} \cup \Sigma_{t_n}(Q_n)$ is obtained by adjoining the new rule $\bar{q}$ to the set $\Sigma_{t_n}(Q_n) = \{\Sigma_{t_n}(q') \mid q' \in Q_n\}$. Finally, setting $s_{n+1} := t_n \circ s_n$ completes the construction of the finite unfolding sequence for G of length $n+1$ based on $m\colon l \hookrightarrow (\lambda_n \circ a)$; the result is the triple $\big\langle \{U_i\}_{i \le n+1}, \{\lambda_i\}_{i \le n+1}, \{t_i\}_{i < n+1} \big\rangle$.

Remark E.2 (Reachable vs. concurrent) This definition does not directly use concurrent objects. The reason is that in this way one can separate the definition of unfoldings from the fact that these unfoldings are actually occurrence grammars. Nevertheless, Lemma E.3 allows to reformulate the unfolding algorithm in a static way as known from the theory of Petri nets (cf. Section 5.2).

This definition of sequences of growing unfoldings is sound in the sense that all grammars of any such chain are actually occurrence grammars.

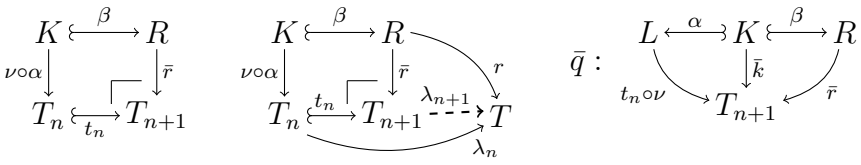
Lemma E.3 (Soundness of unfolding sequences) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, let $G = \langle Q, s\colon S \to T \rangle$ be an $\mathscr{M}$-linear typed grammar, and let $\big\langle \{U_i\}_{i \le n}, \{\lambda_i\}_{i \le n}, \{t_i\}_{i < n} \big\rangle$ be an unfolding sequence.

Then for each $i \in \mathbb{N}$, the grammar U_i is an occurrence grammar.

Proof. The proof is by induction on the length of the sequence. The base case, $n = 0$ is as follows. The only grammar to consider is $U_0 = \langle \varnothing, \mathrm{id}_S\colon S \to S \rangle$. First, as $\mathscr{M}$ contains all isomorphisms, the start object $s_0 = \mathrm{id}_S$ is an $\mathscr{M}$-morphism; all other requirements of occurrence grammars are trivially satisfied, including dissectability.

For the induction step, let $\big\langle \{U_i\}_{i \le n+1}, \{\lambda_i\}_{i \le n+1}, \{t_i\}_{i < n+1} \big\rangle$ be an unfolding sequence; then the induction hypothesis is that for all $i \le n$, the grammar U_i is an occurrence grammar. The proof obligation is to check that also U_{n+1} is an occurrence grammar.

By the definition of unfolding sequence, there is a uniquely determined rule $\bar{q} = (l' \leftarrow k' \rightarrow \bar{r}) \in Q_{n+1}$ which is not contained in T_n, i.e. such that $\bar{r}$ is not included in t_n, in signs $\bar{r} \not\sqsubseteq t_n$. Moreover there is a unique (inclusion) morphism $\nu \colon l' \hookrightarrow t_n$ in $\mathbb{C}\!\downarrow\! T$ such that the situation illustrated by the following diagrams holds.



(21)

Here $\nu = a \circ m$ for a reachable object $a \in \mathbb{C} \downarrow T_n$ and an $\mathscr{M}$-morphism m. By Lemma D.6, $[a]$ is an $\mathscr{M}$-subobject and hence also ν and $\nu \circ \alpha$ are $\mathscr{M}$-morphisms. This in turn implies that U_{n+1} is actually a $\mathbb{C}\!\downarrow\! T_{n+1}$ grammar.

For causal soundness it is enough to show that s_{n+1} is not caused by $\bar{q}$. However $[\bar{k}] = [\bar{r}] \sqcap [t_n]$ and $s_{n+1} \sqsubseteq t_n$ imply $\bar{q} \not\prec s$. Further, causal completeness follows from the fact that $[\nu]$ is a concurrent sub-object w.r.t. U_n and the fact that $[t_n \circ \nu] \sqcap [\bar{r}] = [\bar{k}]$. Moreover type minimality is a direct consequence of the construction of T_{n+1} as the pushout of T_n and R over K, and local finiteness is trivially true.

For backwards determinism, let $q \in Q_{n+1} \setminus \{\bar{q}\}$; then $r_q \sqcap \bar{r} = \bar{k} \sqsubseteq \bar{k} \sqcup k_q$, which again follows from the pushout construction in (21). Further, for acyclicity, $\bar{q}$ is maximal w.r.t. causality and hence cannot be part of a cycle. Moreover, as $[\nu]$ is a concurrent subobject of U_n, there is no cycle in $\nearrow|_{\lfloor t_n \circ \nu \rfloor}$. Using causal maximality of $\bar{q}$ once more, the new rule $\bar{q}$ can be appended to any linearization of $\nearrow|_{\lfloor t_n \circ \nu \rfloor}$ to obtain one of $\nearrow|_{\lfloor q \rfloor}$.

Finally, dissectability is inherited from U_n for all subsets $Q' \subseteq Q_{n+1}$ which do not contain $\bar{q}$. For arbitrary $Q' \subseteq Q_{n+1}$, also $\lfloor Q' \rfloor \setminus \{\bar{q}\}$ is cause closed, and $s \sqcup \bigsqcup_{q' \in \lfloor Q' \rfloor \setminus \{\bar{q}\}} \sqsubseteq t_n$. That also $\bar{r} \sqcup (s \sqcup \bigsqcup_{q' \in \lfloor Q' \rfloor \setminus \{\bar{q}\}})$ is an $\mathscr{M}$-subobject can be shown by pushout splitting of the pushout on the left in (21) at the inclusion morphism from $\bar{k}$ to $s \sqcup \bigsqcup_{q' \in \lfloor Q' \rfloor \setminus \{\bar{q}\}}$. □

To ensure completeness of unfoldings, only sequences with fair choices are used to construct full unfoldings.

☼ Definition E.4 (Fair unfolding chains) Let $\mathbb{C}$ be a weakly $\mathscr{M}$-adhesive category, and let G be an $\mathscr{M}$-linear typed grammar. An *(infinite) unfolding chain for G* is defined as a triple of sequences $\langle \{U_i\}_{i \in \mathbb{N}}, \{\lambda_i\}_{i \in \mathbb{N}}, \{t_i\}_{i \in \mathbb{N}} \rangle$ such that for each $n \in \mathbb{N}_0$ the triple of the truncated sequences $\langle \{U_i\}_{i \leq n}, \{\lambda_i\}_{i \leq n}, \{t_i\}_{i < n} \rangle$

is a finite unfolding sequence for G (or $t_n = \mathrm{id}_{T_n}$).[90] A *fair unfolding chain* is an unfolding chain $\langle \{U_i\}_{i\in\mathbb{N}}, \{\lambda_i\}_{i\in\mathbb{N}}, \{t_i\}_{i\in\mathbb{N}} \rangle$ such that for each $i \in \mathbb{N}$, each rule $q = (l \leftarrow k \rightarrow r) \in Q$ with left hand side $l\colon L \to T$, each concurrent $\mathscr{M}$-subobject $l'\colon L \hookleftarrow T_i$ there is a natural number $j \geq i$ and a rule occurrence $q_j = (l_j \hookleftarrow k_j \hookrightarrow r_j) \in Q_j$ such that $\Sigma_{\lambda_j}(q_j) = q$ and $l_j = t_i^j \circ l'$ where the morphism $t_i^j\colon T_i \hookleftarrow T_j$ is the composition of the chain $\{t_n\colon T_n \hookleftarrow T_{n+1}\}_{i\leq n<j}$. ☼

Fair unfolding sequences will ensure completeness of full unfoldings, at least for consuming grammars.

Lemma E.5 (Completeness of fair unfolding chains) Let $G = \langle Q, s\colon S \to T\rangle$ be a *consuming* T-typed grammar and let $\langle \{U_i\}_{i\in\mathbb{N}}, \{\lambda_i\}_{i\in\mathbb{N}}, \{t_i\}_{i\in\mathbb{N}} \rangle$ be a fair unfolding chain. Then for each derivation $\mathfrak{X}\colon s \Rightarrow\!\!\!\!\!\!\cdot\;^{Q} a$ in G there is a (minimal) natural number $i \in \mathbb{N}$ such that there is a derivation $\mathfrak{Z}\colon s_i \Rightarrow\!\!\!\!\!\!\cdot\;^{Q_i} a_i$ in U_i which is mapped to $\mathfrak{X}$ via λ_i, i.e. such that $\mathfrak{X} = \Sigma_{\lambda_i} \circ \mathfrak{Z}$.

Proof. The proof is by induction on the length of the derivation $\mathfrak{X}\colon s \Rightarrow\!\!\!\!\!\!\cdot\;^{Q} a$ in the grammar G. The base case of the induction is trivially satisfied, as $\lambda_0\colon S \to T$ is equal to $s\colon S \to T$.

For the induction step, let

$$\mathfrak{X} \fatsemi \mathfrak{X}\colon s \Rightarrow\!\!\!\!\!\!\cdot\;^{Q'} b \Rightarrow\!\!\!\!\!\!\cdot\;^{\langle q,m\rangle} a$$

be a derivation in the grammar G. Using the induction hypothesis there is a natural number $j \in \mathbb{N}$ and a derivation $\mathfrak{Z}'\colon s_j \Rightarrow\!\!\!\!\!\!\cdot\;^{Q_j} b_j$ in U_j such that $\Sigma_j \circ \mathfrak{Z}' = \mathfrak{X}$. Now $m\colon l_q \hookrightarrow b$ induces a concurrent subobject $b_j \circ m\colon L_q \hookrightarrow T_j$ of U_j. By the definition of fair unfolding chains, there must be some $i > j$ such that there is some $q' \in Q_i$ such that both $b \circ m = l_{q'}$ and $q = \Sigma_{\lambda_i} \circ q'$ hold. Using the properties of reached objects (see Lemma D.6), in the derivation $\mathfrak{Z} := \Sigma_{t_j^i} \circ \mathfrak{Z}'\colon s_i \Rightarrow\!\!\!\!\!\!\cdot\;^{Q_i} t_j^i \circ b_j$, the rule $q' \in Q_i$ has not been used because of $q' \in \ulcorner (t_j^i \circ b_j)\urcorner$ where $t_j^i\colon T_j \hookleftarrow T_i$ is the composition of the chain of type object inclusions $\{t_n\colon T_n \hookleftarrow T_{n+1}\}_{i\leq n<j}$. Applying the rule q' at the object $t_j^i \circ b_j$ yields a direct derivation $\mathfrak{Z}\colon t_j^i \circ b_j \Rightarrow\!\!\!\!\!\!\cdot\;^{\langle q',m\rangle} a_i$ such that $\Sigma_{\lambda_i} \circ \mathfrak{Z} = \mathfrak{X}\colon b \Rightarrow\!\!\!\!\!\!\cdot\;^{\langle q,m\rangle} a$. By Combining the above, $\mathfrak{X} \fatsemi \mathfrak{X} = \Sigma_{\lambda_i} \circ (\mathfrak{Z} \fatsemi \mathfrak{Z})$ is established. □

Before addressing the issue of existence of fair unfolding sequences, the next definition assumes that they exist.

[90] The latter case is added to include the case in which the full behaviour of the original grammar is finite.

☼ Definition E.6 (Full unfolding) Let $G = \langle Q, s\colon S \to T\rangle$ be an $\mathscr{M}$-linear consuming grammar and let $\big\langle \{U_i\}_{i\in\mathbb{N}}, \{\lambda_i\}_{i\in\mathbb{N}}, \{t_i\}_{i\in\mathbb{N}}\big\rangle$ be a fair unfolding chain. Then the *full unfolding of* G is obtained by taking the colimit $\{t'_i\colon T_i \hookleftarrow T'\}_{i\in\mathbb{N}}$ of the chain $\{t_i\colon T_i \hookleftarrow T_{i+1}\}_{i\in\mathbb{N}}$ in $\mathbb{C}$; this yields the grammar

$$U = \left\langle \bigcup_{i\in\mathbb{N}} \Sigma_{t'_i} \circ Q_i,\ S \leftarrow t'_0 \rightarrow T \right\rangle$$

with a unique *folding morphism* $\lambda\colon T' \to T$ which satisfies the equation $\lambda_i = \lambda \circ t'_i\colon T_i \hookleftarrow T$ for all $i \in \mathbb{N}$. ☼

In fact, this construction yields an occurrence grammar.

◌ Lemma E.7 (Soundness of the full unfolding) The full unfolding of any consuming, $\mathscr{M}$-linear grammar is an occurrence grammar.

Proof. The main idea is almost all properties are inherited from the finite (partial) unfoldings. The only property that is not inherited from the finite sub-grammars is type minimality. The latter follows from Proposition 7.19. applied to the chain $\{t'_n \sqsubseteq t'_{n+1}\}_{n\in\mathbb{N}}$. □

A sufficient condition for the existence of fair unfolding sequences is finite or countable "size" of all components of the original grammar. Here the size of an object $A \in \mathbb{C}$ is identified with the cardinality of the subobject lattice $\mathrm{Sub}_{\mathscr{M}}(A)$.

◌ Lemma E.8 (Existence of fair unfolding chains) Let $G = \langle Q, s\colon S \to T\rangle$ be an occurrence grammar. Then there exists a fair unfolding chain for G if the set of rules Q is countable, the subobject lattice $\mathrm{Sub}_{\mathscr{M}}(s)$ is countable and also $\mathrm{Sub}_{\mathscr{M}}(r_q)$ is countable for each rule $q = (l_q \leftarrow k_q \rightarrow r_q) \in Q$.

Proof. It is enough to show that in the grammar U_n of any finite unfolding sequence $\big\langle \{U_i\}_{i\le n}, \{\lambda_i\}_{i\le n}, \{t_i\}_{i<n}\big\rangle$, there are only countably many choices for a concurrent subobject of the form $\nu\colon L_q \hookleftarrow T_n$ for a rule $q \in Q$.

In particular, the subobject lattice $\mathrm{Sub}(T_n)$ is countable, which can be shown by induction using distributivity. The main observation is that the cardinality of $\mathrm{Sub}(T_{n+1})$ is at most as big as the product of the cardinalities of $\mathrm{Sub}(T_n)$ and $\mathrm{Sub}(R_q)$ where $r_q\colon R_q \to T$ is the right hand side of the rule in Q for which a new rule occurrence has been added to U_{n+1}. □

One last fact about unfoldings that follows directly by construction is the irredundancy of unfoldings.

Lemma E.9 (Irredundancy of unfoldings) Let $G = \langle Q, s\colon S \to T\rangle$ be an $\mathscr{M}$-linear, consuming grammar, and let $U = \langle Q', s'\colon S \to T'\rangle$ be its unfolding with folding morphism $\lambda\colon T' \to T$.

Then for each concurrent subobject $\nu\colon L \hookrightarrow T'$ such that $\lambda \circ \nu = l_q$ for some rule $q \in Q$, there is exactly one rule $q' \in Q'$ such that $q = \Sigma_\lambda(q')$ and $\lambda_{q'} \circ l_{q'} = l_q$.

Proof. Existence follows from fairness of unfolding chains or completeness of the unfolding. Uniqueness follows from the requirement, that the rule occurrences that are added in unfolding sequences are required to be new (cf. Definition E.1). □

E.1 Retyping functors and grammar morphisms

The two main technical facts about retyping functors is that they form a category, and that they preserve all relevant colimits in weakly $\mathscr{M}\omega$-adhesive categories.

Lemma E.10 (Retyping functors via cartesian transformations) Any functor $\mathcal{F}\colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow Y$ between slice categories is a retyping functor if and only if there exists a cartesian natural transformation $\mathbb{r}\colon |\_|_Y \circ \mathcal{F} \to |\_|_X$.

Proof. If $\mathcal{F}$ is a retyping functor, then it is of the form $\Sigma_f \circ h^*$ for some $\mathbb{C}$-span $X \leftarrow h - V - f \rightarrow Y$. Now the co-unit $\varepsilon\colon \Sigma_f \circ f^* \to \mathrm{id}_{\mathbb{C}\downarrow X}$ of the adjunction $\Sigma_f \dashv f^*$ induces a cartesian transformation $\tilde{\varepsilon} := \{|\varepsilon_a|_X\colon |\mathcal{F}(a)|_Y \to |a|_X\}$ from $|\_|_Y \circ \mathcal{F}$ to $|\_|_X$.

Conversely, given a cartesian transformation $\mathbb{r}\colon |\_|_Y \circ \mathcal{F} \to |\_|_X$, the functor $\mathcal{F}$ acts by pulling back along $\mathbb{r}_{\mathrm{id}_Y}$ followed by composition with $\mathcal{F}(\mathrm{id}_Y)$. The pullback functor $\mathbb{r}^*_{\mathrm{id}_Y}$ maps $a \in \mathbb{C}\downarrow X$ to $\mathbb{r}_{!_a}$ and $\psi\colon a \to b$ to $\mathcal{F}(\psi)$. □

Corollary E.11 (Category of retyping functors) Given a category $\mathbb{C}$, retyping functors between slice categories of $\mathbb{C}$ form a category where identities and compositions are inherited from $\mathbb{C}\mathbb{a}\mathbb{t}$.

Proof. As a direct consequence of the Pullback Lemma, cartesian transformations are closed under vertical and horizontal composition, and the identity natural transformation on a functor is cartesian. Hence the composition of two retyping functors is again a retyping functor. More precisely, as illustrated in Figure 113, given retyping functors $\mathcal{F}\colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow Y$ and $\mathcal{G}\colon \mathbb{C}\downarrow Y \to \mathbb{C}\downarrow Z$ with witnessing cartesian transformations $\mathbb{r}\colon |\_|_Y \circ \mathcal{F} \to |\_|_X$ and $\mathbb{o}\colon |\_|_Z \circ \mathcal{G} \to |\_|_Y$,

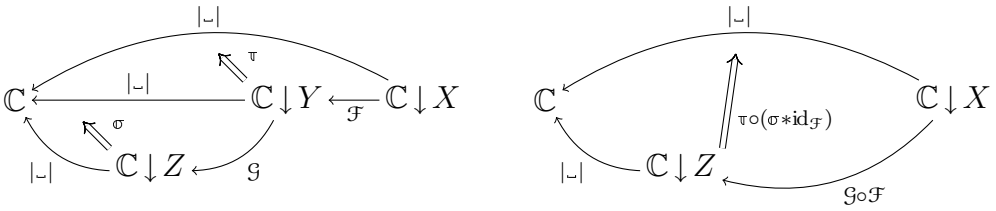


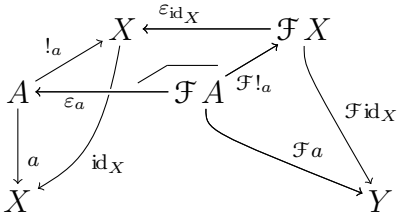
Figure 113: Composition of retyping functors between slice categories

then $\mathbb{T} \circ (\mathfrak{S} * \mathrm{id}_{\mathcal{F}})$ is a cartesian transformation with domain $|\_|_Z \circ \mathcal{G} \circ \mathcal{F}$ and codomain $|\_|_X$. Moreover, the identity on a slice category is a retyping functor. □

Lemma E.12 (Determination of retyping functors) Let $\mathcal{F} \colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow Y$ be a retyping functor. Then $\mathcal{F}$ is fully determined by $|\_|_Y \circ \mathcal{F}$ and $\mathcal{F}(\mathrm{id}_X)$.

Proof. Let $\mathcal{G} \colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow Y$ be a retyping functor with $|\_|_Y \circ \mathcal{G} = |\_|_Y \circ \mathcal{F}$ and $\mathcal{G}(\mathrm{id}_X) = \mathcal{F}(\mathrm{id}_X)$. Now derive the following in $\mathbb{C}$.

$$\begin{aligned} \mathcal{G}(a) &= \mathcal{G}(\mathrm{id}_X) \circ |\mathcal{G}(!_a)|_Y \\ &= \mathcal{F}(\mathrm{id}_A) \circ |\mathcal{F}(!_a)|_B \\ &= \mathcal{F}(a) \end{aligned}$$

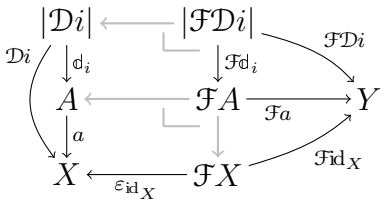


Hence $\mathcal{F}$ and $\mathcal{G}$ coincide on objects; however they also coincide on arrows because of $|\_|_B \circ \mathcal{G} = |\_|_B \circ \mathcal{F}$. □

Lemma E.13 (Retyping universal colimits) Let $\mathcal{F} \colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow Y$ be a retyping functor, and further let $\mathcal{D} : \mathbb{I} \to (\mathbb{C}\downarrow X)$ be a diagram with colimit $\mathbb{d} \colon \mathcal{D} \to \Delta_{(A \text{–} a \to X)}$ such that $\mathrm{id}_{|\_|_X} * \mathbb{d} \colon |\_|_X \circ \mathcal{D} \to \Delta_{(A)}$ is a universal colimit of $|\_|_X \circ \mathcal{D}$.

Then $\mathcal{F}(a)$ is determined by $|\_|_Y \circ \mathcal{F}$ and the family $\{\mathcal{F} \circ \mathcal{D}(i)\}_{i \in \mathbb{I}}$.

Proof. Let $\mathcal{G} \colon \mathbb{C}\downarrow X \to \mathbb{C}\downarrow Y$ be another retyping functor that satisfies the equations $|\_|_Y \circ \mathcal{G} = |\_|_Y \circ \mathcal{F}$ and $\mathcal{G}(\mathcal{D}(i)) = \mathcal{F}(\mathcal{D}(i))$ for all $i \in \mathbb{I}$. The proof obligation is to establish the equation $\mathcal{G}(a) = \mathcal{F}(a)$.



The main observation is that $\mathcal{F}(a)$ is exactly the unique morphism that satisfies the equation $\mathcal{F}(\mathcal{D}(i)) = \mathcal{F}(a) \circ |\mathcal{F}(\mathbb{d}_i)|$ for all $i \in \mathbb{I}$. This is a direct consequence of the universality of the colimit $\mathrm{id}_{|\llcorner|_X} * \mathbb{d}$. However, for the same reason, also $\mathcal{G}(\mathcal{D}(i)) = \mathcal{G}(a) \circ |\mathcal{G}(\mathbb{d}_i)|$ holds for all $i \in \mathbb{I}$. Finally, using the assumptions about $\mathcal{G}$, derive $\mathcal{F}(\mathcal{D}(i)) = \mathcal{G}(a) \circ |\mathcal{F}(\mathbb{d}_i)|$ for arbitrary $i \in \mathbb{I}$, which implies that $\mathcal{F}(a) = \mathcal{G}(a)$. □

Remark E.14 All proofs that involve the application of grammar morphisms to rules do not treat the trivial cases in which the rules are mapped to identity rules. Each of these trivial cases can be easily verified.

Proposition E.15 (Local determination of occurrence grammars) Let $O = \langle Q', s' \colon S' \hookrightarrow T' \rangle$ be an occurrence grammar, let $G = \langle Q, s \colon S \to T \rangle$ be any grammar, and let $\mathcal{F} \colon O \to G$ be a grammar morphism. Then $\mathcal{F}$ is fully determined by $|\llcorner|_{T'} \circ \mathcal{F}$, and the family $\{\mathcal{F}(r_q)\}_{q \in Q'}$.

Proof. Let $\{q_i \mid i \in \mathbb{N}\} = Q'$ be an enumeration of the rules in Q' which is compatible with causality, i.e. $j > i$ implies $q_j \not\preceq q_i$. Now, for each $n \in \mathbb{N}$, define $[w_n] := s \sqcup \bigsqcup_{0<i\leq n} r_{q_i}$, which is an $\mathscr{M}$-subobject (by dissectability); further let $m_n \colon w_n \hookrightarrow w_{n+1}$ in $\mathbb{C}\!\downarrow\! T'$ be the unique "inclusion morphism".

First, the following induction on n establishes that $\mathcal{F}(w_n)$ is fully determined by $|\llcorner|_{T'} \circ \mathcal{F}$ and the family $\{\mathcal{F}(r_{q_i})\}_{i \leq n}$. In each case, let $\mathcal{G} \colon O \to G$ be a grammar morphism satisfying $|\llcorner|_{T'} \circ \mathcal{F} = |\llcorner|_{T'} \circ \mathcal{G}$ and $\mathcal{F}(r_{q_i}) = \mathcal{G}(r_{q_i})$ for all $i \leq n$.

$n = 0$: The grammar morphism $\mathcal{G} \colon O \to G$ must satisfy $\mathcal{G}(s') = s = \mathcal{F}(s')$. As $[w_0] = [s]$, there is an isomorphism $\iota \colon w_0 \xrightarrow{\cong} s$ for which

$$\begin{aligned} \mathcal{F}(w_0) &= \mathcal{F}(s) \circ |\mathcal{F}(\iota)| \\ &= \mathcal{G}(s) \circ |\mathcal{G}(\iota)| \\ &= \mathcal{G}(w_0) \end{aligned}$$

holds.[91]

$n \rightsquigarrow n+1$: Let $q_{n+1} = (l \leftarrow\!\alpha\!- k \!-\!\beta\!\rightarrow r)$ be the $n+1$-st rule with "inclusion morphism" $\psi \colon k \hookrightarrow w_n$ and $\zeta \colon r \hookrightarrow w_{n+1}$ in $\mathbb{C}\!\downarrow\! T'$. Then $w_n \hookleftarrow m_n \hookrightarrow w_{n+1} \hookleftarrow \zeta \hookrightarrow r$ is the pushout of $w_n \leftarrow\!\psi\!\rightarrow k \leftarrow\!\beta\!\rightarrow r$ in $\mathbb{C}\!\downarrow\! T'$, yielding a pushout square $\begin{smallmatrix} k \\ w_n \end{smallmatrix} \downarrow \overrightarrow{\lrcorner} \downarrow \begin{smallmatrix} r \\ w_{n+1} \end{smallmatrix}$. By the induction hypothesis, $\mathcal{F}(w_n) = \mathcal{G}(w_n)$ and further

[91] One could also have assumed w.l.o.g. that $w_0 = s$.

$\mathcal{F}(r) = \mathcal{G}(r)$ by assumption. As a consequence, using weak $\mathscr{M}$-adhesivity, Lemma E.13 is applicable and implies that $\mathcal{F}(w_{n+1}) = \mathcal{G}(w_{n+1})$.

Now, using minimality of the type T', the family $\{!_{w_n} : w_n \hookleftarrow \mathrm{id}_{T'}\}_{n\in\mathbb{N}}$ is actually a colimit of the chain $\{m_n : w_n \hookleftarrow w_{n+1}\}_{n\in\mathbb{N}}$; and this colimit is universal by weak $\mathscr{M}\omega$-adhesivity. Hence Lemma E.13 can be applied once more to obtain that $\mathcal{F}(\mathrm{id}_{T'}) = \mathcal{G}(\mathrm{id}_{T'})$. Finally Lemma E.12 yields $\mathcal{F} = \mathcal{G}$. □

E.2 Coreflection theorem

Finally all building blocks for the proof of the Coreflection Theorem are available. During the proof it will be convenient to use sub-occurrence grammars.

☼ Definition E.16 (Sub-occurrence grammar) Let $O = \langle Q^\star, s^\star : S^\star \to T^\star\rangle$ be an occurrence grammar and $Q' \subseteq Q^\star$ a cause-closed set of rules, i.e. the equation $Q' = \lfloor Q' \rfloor$ must hold. Further let $[w] = [s^\star] \sqcup \bigsqcup_{q'\in Q'} [r_{q'}]$ be the join of $s^\star$ and all right hand sides in Q'.

Then $O\restriction_w = \langle Q\restriction_w, S^\star \hookleftarrow s\restriction_w \to |w|_{T^\star}\rangle$ is the unique occurrence grammar such that $s^\star = w \circ s\restriction_w$ and $Q' = \Sigma_w(Q\restriction_w)$, i.e. $\Sigma_w : O\restriction_w \to O$ is the canonical embedding. More explicitly, $s\restriction_w : S^\star \hookleftarrow |w|_{T^\star}$ is the unique $\mathscr{M}$-morphism which witnesses the inclusion $s^\star \sqsubseteq w$, and for any rule $q = (l \leftarrow k \to r) \in Q'$ there is a unique rule $q\restriction_w = (l\restriction_w \leftarrow k\restriction_w \to r\restriction_w) \in Q\restriction_w$ such that $l\restriction_w$, $k\restriction_w$ and $r\restriction_w$ are $\mathscr{M}$-morphisms which witness the inclusions $l \sqsubseteq w$, $k \sqsubseteq w$, $r \sqsubseteq w$, respectively. ☼

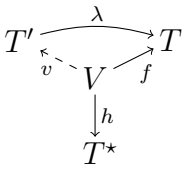
☼ Theorem E.17 (Coreflection) Let $G = \langle Q, s : S \to T\rangle$ be an $\mathscr{M}$-linear, consuming grammar in some weakly $\mathscr{M}\omega$-adhesive category $\mathbb{C}$. Further let $U = \langle Q', s' : S \hookleftarrow T'\rangle$ be an unfolding of G with folding morphism $\lambda : T' \to T$.

Then for any occurrence grammar $O = \langle Q^\star, s^\star : S^\star \to T^\star\rangle$ and any grammar morphism $\mathcal{F} : O \to G$ there is a unique grammar morphism $\mathcal{H} : O \to U$ which factors through Σ_λ, i.e. such that $\mathcal{F} = \Sigma_\lambda \circ \mathcal{H}$.

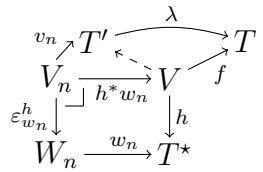
Proof. As a first observation, any morphism $\mathcal{H}$ which satisfies $\mathcal{F} = \Sigma_\lambda \circ \mathcal{H}$ must satisfy $|\_|_T \circ \mathcal{F} = |\_|_T \circ \Sigma_\lambda \circ \mathcal{H} = |\_|_{T'} \circ \mathcal{H}$. In other words, the grammar morphism $\mathcal{F}$ prescribes the action of $\mathcal{H}$ on morphisms, and moreover, for any $a \in \mathbb{C}{\downarrow}T^\star$, also the domain of $\mathcal{H}(a)$ is specified. Hence, by Lemma E.12, as will be elaborated on below, it is enough to find a "suitable" arrow $v : |\mathcal{F}(\mathrm{id}_{T^\star})| \to T'$ as the image of $\mathrm{id}_{T^\star}$ and show that there is only one such choice.

Starting with the actual proof, begin as in Proposition E.15, and let $\{q_i \mid i \in \mathbb{N}\} = Q^\star$ be an enumeration of the rules in $Q^\star$ which is compatible with causality, i.e. $j > i$ implies $q_j \not\preceq q_i$. Now, for each $n \in \mathbb{N}$, define the join

$[w_n] := s \sqcup \bigsqcup_{0<i\leq n} r_{q_i}$, which is an $\mathscr{M}$-subobject (by dissectability) with $w_0 = s$; further let $m_n \colon w_n \hookrightarrow w_{n+1}$ in $\mathbb{C}\downarrow T^\star$ be the unique "inclusion morphism". In this way, $\mathrm{id}_{T^\star}$ arises as the colimit of the chain $\mathcal{W} = \{m_n \colon w_n \hookrightarrow w_{n+1}\}_{n\in\mathbb{N}}$, i.e. $\mathbb{w} := \{!_{w_n} \colon w_n \hookrightarrow \mathrm{id}_{T^\star}\}_{n\in\mathbb{N}}$ is a colimit of $\mathcal{W}$ by type minimality.

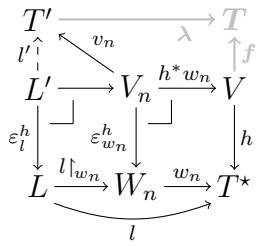


Next, for the existence of some suitable $\mathcal{H} \colon O \to G$, factor $\mathcal{F}$ as $\Sigma_f \circ h^*$ for some $\mathbb{C}$-span $T^\star \leftarrow h - V - f \rightarrow T$. By weak $\mathscr{M}\omega$-adhesivity, the colimit $|\_|_{T^\star} * \mathbb{w}$ is universal, and hence $\{h^*(w_n)\}_{n\in\mathbb{N}}$ is a colimit of $|\_|_V \circ h^* \circ \mathcal{W}$, which is the same $\mathbb{C}$-diagram as $|\_|_T \circ \mathcal{F} \circ \mathcal{W}$. Now, the main idea is to construct for each $n \in \mathbb{N}$ the "image" v_n of w_n, i.e. $v_n \colon |h^*(w_n)| \to T'$. Then there arises a mediating morphism $v \colon V \to T'$ and $\mathcal{H} := \Sigma_v \circ h^*$ will satisfy $\mathcal{F} = \Sigma_\lambda \circ \mathcal{H}$.



Moreover each of these v_n will induce simulation strategies of $O\restriction_{w_n}$ such that all derivations that only use rules $q_1, \ldots, q_n \in Q^\star$ which are contained in w_n have a fixed counterpart in U. More precisely, let ε^h be the counit of the adjunction $\Sigma_h \dashv h^*$, then $\mathcal{H}_n := \Sigma_{v_n} \circ |\varepsilon^h_{w_n}|^*$ is a grammar morphism from $O\restriction_{w_n}$ to U, such that $\Sigma_\lambda \circ \mathcal{H}_n = \mathcal{F} \circ \Sigma_{w_n}$.

For $n = 0$, the value is $v_0 = s' \colon S \hookrightarrow T'$. For the inductive step, take as induction hypothesis that there are values $v_0, \ldots, v_n$ such that the equation $v_j = v_{j+1} \circ |h^*(m_j)|$ holds for all $0 \leq j < n$ (or $v_j = v_{j+1} \circ |\mathcal{F}(m_j)|$, which is equivalent), and moreover $\mathcal{H}_n := \Sigma_{v_n} \circ |\varepsilon^h_{w_n}|^*$ satisfies $\Sigma_\lambda \circ \mathcal{H}_n = \mathcal{F} \circ \Sigma_{w_n}$.



Now let $q_{n+1} = (l \leftarrow \alpha - k - \beta \rightarrow r)$ be the $n{+}1$-st rule. Then $[l] \in \mathrm{Sub}_{\mathscr{M}}(T^\star)$ is a concurrent sub-object which is contained in w_n via a unique "inclusion morphism" $l_n \colon l \hookrightarrow w_n$. As a result, $l\restriction_{w_n} := |l_n| \in \mathbb{C}\downarrow W_n$ is a concurrent subobject of $O\restriction_{w_n}$, which corresponds to a concurrent subobject $[\mathcal{H}_n(l\restriction_{w_n})]$ in U as illustrated. As U is constructed from a fair unfolding sequence, there is a rule $q' = (l' \leftarrow k' \rightarrow r') \in Q'$ such that $l' = \mathcal{H}_n(l\restriction_{w_n})$ and $\Sigma_\lambda(q') = \mathcal{F}(q_{n+1})$.

Now, all preparations are made to "extend" v_n to w_n using r' as image of r. For this let $\psi \colon k \hookrightarrow w_n$ and $\zeta \colon r \hookrightarrow w_{n+1}$ in $\mathbb{C}\downarrow T^\star$ be the inclusion "inclusion morphism". Then the cospan $w_n \hookleftarrow m_n \rightarrow w_{n+1} \hookleftarrow \zeta \rightarrowtail r$ is the pushout of $w_n \hookleftarrow \psi \rightarrowtail k \hookleftarrow \beta \rightarrow r$ in $\mathbb{C}\downarrow T^\star$, yielding a pushout square $\begin{smallmatrix} k \\ w_n \end{smallmatrix} \downarrow \overrightarrow{\lrcorner} \downarrow \begin{smallmatrix} r \\ w_{n+1} \end{smallmatrix}$. Moreover, the co-span $|\mathcal{F}(w_n)| \hookleftarrow |\mathcal{F}(m_n)| \rightarrow |\mathcal{F}(w_{n+1})| \hookleftarrow \mathcal{F}(\zeta) \rightarrowtail |\mathcal{F}(r)|$ is a pushout of the span $|\mathcal{F}(w_n)| \hookleftarrow |\mathcal{F}(\psi)| \rightarrowtail |\mathcal{F}(k)| \hookleftarrow |\mathcal{F}(\beta)| \rightarrow |\mathcal{F}(r)|$ since pushouts of pairs of $\mathscr{M}$-morphisms are universal by weak $\mathscr{M}$-adhesivity (see Figure 114). Finally, v_{n+1} is defined as the unique morphism $v_{n+1} \colon |\mathcal{F}(w_{n+1})| \to T'$ which satisfies $v_n = v_{n+1} \circ |\mathcal{F}(m_n)|$

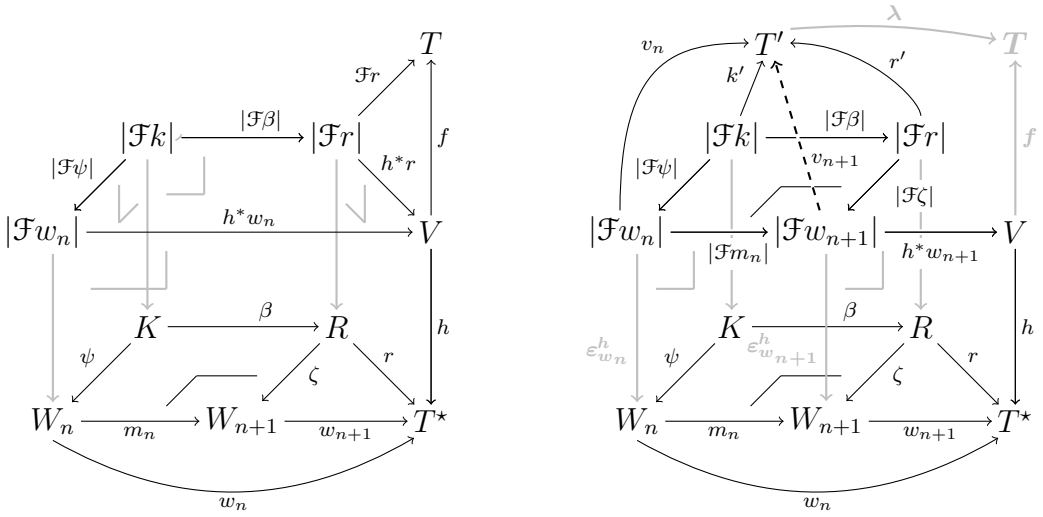


Figure 114: Constructing the image of the chain $m_n \colon w_n \hookrightarrow w_{n+1}$

and $r' = v_{n+1} \circ |\mathcal{F}(\zeta)|$ (see the right hand diagram in Figure 114); using diagram chasing arguments in the latter diagram, conclude that $\mathcal{H}_{n+1} := \Sigma_{v_{n+1}} \circ |\varepsilon^h_{w_{n+1}}|^*$ satisfies the equation $\Sigma_\lambda \circ \mathcal{H}_{n+1} = \mathcal{F} \circ \Sigma_{w_{n+1}}$.

Thus, using universality of colimits of ω-chains of $\mathscr{M}$-morphisms, the family $\{h^*(w_n)\}_{n\in\mathbb{N}}$ is a colimit of the chain $\{|\mathcal{F}(m_n)| \colon |\mathcal{F}(w_n)| \hookrightarrow |\mathcal{F}(w_{n+1})|\}_{n\in\mathbb{N}}$. Further $v_n = v_{n+1} \circ |\mathcal{F}(m_n)|$ holds for all $n \in \mathbb{N}$. Hence there exists a unique mediating morphism $v \colon V \to T'$ satisfying $v_n = v \circ h^*(w_n)$ for all $n \in \mathbb{N}$. Moreover, $\lambda \circ v = f$, and hence $\mathcal{H} = \Sigma_v \circ h^*$ satisfies $\Sigma_\lambda \circ \mathcal{H} = \mathcal{F}$. As an immediate consequence of the above constructions, $\mathcal{H}$ is actually a grammar morphism.

The final part of the proof concerns uniqueness of the grammar morphism $\mathcal{H}$. Let $\mathcal{E} \colon O \to G$ be another grammar morphism for which the equation $\mathcal{F} = \Sigma_\lambda \circ \mathcal{E}$ holds. Since this already implies that $|\_|_{T'} \circ \mathcal{E} = |\_|_{T'} \circ \mathcal{H}$, use Proposition E.15 to conclude that it suffices to show $\mathcal{E}(r) = \mathcal{H}(r)$ for all right hand sides of rules $q = (l \leftarrow k \rightarrow r) \in Q^\star$.

Now, the equation $\mathcal{E}(r) = \mathcal{H}(r)$ will be verified by induction on the depth of each rule $q = (l \leftarrow k \rightarrow r) \in Q^\star$, i.e. by induction on the maximal length of all causal chains $q_1 \prec \cdots \prec q_n = q$ below q in O.

$n = 0$ This means that the inclusion $l \sqsubseteq s^\star$ holds, which it witnessed by a unique inclusion morphism $m \colon l \hookrightarrow s^\star$. Both $\mathcal{E}$ and $\mathcal{H}$ must map l to $s' \circ |\mathcal{F}(m)| =: l'$, which yields a concurrent subobject l'. Now, by

the irredundancy of the unfolding, $\mathcal{E}(q) = \mathcal{H}(q)$ is the unique rule $q' = (l' \leftarrow k' \rightarrow r') \in Q'$ such that $\Sigma_\lambda(q') = \mathcal{F}(q)$.

$n \rightsquigarrow n+1$: By causal completeness, the inclusion $l \sqsubseteq \left(s \bigsqcup_{\tilde{q} \in \lfloor l \rfloor} r_{\tilde{q}}\right) = [w_q]$ holds, which is witnessed by a unique $\mathscr{M}$-morphism $l\restriction_{w_q} : |l| \hookrightarrow |w_q|$. By the induction hypothesis, $\mathcal{E}(r_{\tilde{q}}) = \mathcal{H}(r_{\tilde{q}})$ holds for all $\tilde{q} \in \lfloor l \rfloor$. Using Proposition E.15 on the sub-grammar $O\restriction_{w_q} \xrightarrow{-\Sigma_{w_n}} O$, yields the equation $\mathcal{E} \circ \Sigma_{w_q} = \mathcal{H} \circ \Sigma_{w_q}$; moreover $\mathcal{E} \circ \Sigma_{w_q}(l\restriction_{w_q}) = \mathcal{H} \circ \Sigma_{w_q}(l\restriction_{w_q}) =: l'$ describes a concurrent subobject $[l']$ in U. Further, using irredundancy, conclude that there is a unique rule $q' = (l' \leftarrow k' \rightarrow r') \in Q'$ such that $\Sigma_\lambda(q') = \mathcal{F}(q)$. Hence, necessarily $\mathcal{H}(q) = q' = \mathcal{E}(q)$, and in particular $\mathcal{H}(r) = \mathcal{E}(r)$.

□