Chapter 1
Amalgams, Colimits, and Conceptual Blending

Félix Bou, Enric Plaza, and Marco Schorlemmer

Abstract This chapter is a theoretical exploration of Joseph Goguen's category-theoretic model of conceptual blending and presents an alternative proposal to model blending as *amalgams*, which were originally proposed as a method for knowledge transfer in case-based reasoning. The chapter concludes with a generalisation of the amalgam-based model by relating it to the notion of colimit, thus providing a category-theoretic characterisation of amalgams that is ultimately computationally realisable.

1.1 Introduction

The notion of *amalgam* in a lattice of generalisations was developed in the framework of modelling analogical inference, and case amalgamation in case-based reasoning (CBR) (Ontañón and Plaza, 2010). Case amalgamation models the process of combining two different cases into a new *blended* case to be used in the CBR problem-solving process. As such, the notion of amalgam seems related to but not identical to the notions of *conceptual blending*, also known as conceptual integration (Fauconnier and Turner, 1998). These related notions have in common that there is some combination or fusion of two different sources into a new entity that encompasses selected parts of the sources, but they differ in the assumptions on the entities upon which they work: amalgams work on *cases* (expressed as terms in some language), while conceptual blending works on *mental spaces*.

Félix Bou
Department of Philosophy, University of Valencia, Avda. Blasco Ibáñez 30, 46010 Valencia, Spain
e-mail: `felix.bou@uv.es`

Enric Plaza · Marco Schorlemmer
Artificial Intelligence Research Institute (IIIA-CSIC), C/ Can Planes 2, Campus UAB, 08193 Bellaterra (Cerdanyola del Vallès), Spain
e-mail: `{enric,marco}@iiia.csic.es`

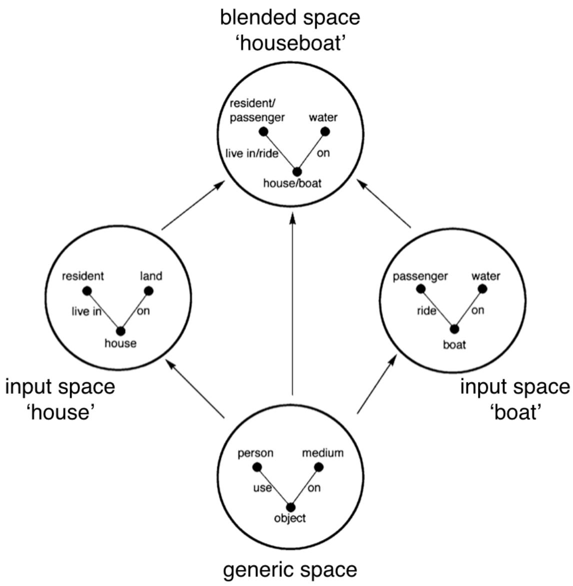


Fig. 1.1: 'Houseboat' blend, adapted from Goguen and Harrell (2010)

Fauconnier and Turner proposed conceptual blending as the fundamental cognitive operation underlying much of everyday thought and language. They model it as a process by which people subconsciously combine particular elements and their relations of originally separate input mental spaces—which do, however, share some common structure modelled as a generic space—into a blended space, in which new elements and relations emerge, and new inferences can be drawn. For instance, a 'houseboat' or a 'boathouse' are not simply the intersection of the concepts of 'house' and 'boat'. Instead, the concepts 'houseboat' and 'boathouse' selectively integrate different aspects of the source concepts in order to produce two new concepts, each with its own distinct internal structure (see Figure 1.1 for the 'houseboat' blend).

Although the cognitive, psychological and neural basis of conceptual blending has been extensively studied (Fauconnier and Turner, 2002; Gibbs, Jr., 2000; Baron and Osherson, 2011) and Fauconnier and Turner's theory has been successfully applied for describing existing blends of ideas and concepts in a varied number of fields, such as linguistics, music theory, poetics, mathematics, theory of art, political science, discourse analysis, philosophy, anthropology, and the study of gesture and of material culture, their theory has been used only in a more constrained way for implementing creative computational systems. Since Fauconnier and Turner did not aim at computer models of cognition, they did not develop the sufficient details for conceptual blending to be captured algorithmically.

Nevertheless, a number of researchers in the field of computational creativity have recognised the potential value of Fauconnier and Turner's theory for guiding the implementation of creative systems, and some computational accounts of conceptual blending have already been proposed (Veale and O'Donoghue, 2000; Pereira, 2007; Goguen and Harrell, 2010; Thagard and Stewart, 2011). They attempt to concretise some of Fauconnier and Turner's insights, and the resulting systems have shown interesting and promising results in creative domains such as interface design, narrative style, poetry generation, or visual patterns. All of these accounts, however, are customised realisations of conceptual blending, which are strongly dependent on hand-crafted representations of domain-specific knowledge, and are limited to very specific forms of blending. The major obstacle for a general account of computational conceptual blending is currently the lack of a mathematically precise theory that is suitable for the rigorous development of creative systems based on conceptual blending.

The only attempt so far to provide a general and mathematically precise account of conceptual blending has been put forward by Goguen, initially as part of algebraic semiotics (Goguen, 1999), and later in the context of a wider theory of concepts that he named Unified Concept Theory (UCT) (Goguen, 2005a); he has also shown its aptness for formalising information integration (Goguen, 2005b) and reasoning about space and time (Goguen, 2006). As it stands, Goguen's account is still very abstract and lacks concrete algorithmic descriptions. There are several reasons, though, that make it an appropriate candidate theory on which to ground the formal model we are aiming at:

- It is an important contribution towards the unification of several formal theories of concepts, including the geometrical conceptual spaces of Gärdenfors (2004), the symbolic conceptual spaces of Fauconnier (1994), the information flow of Barwise and Seligman (1997), the formal concept analysis of Ganter and Wille (1999), and the lattice of theories of Sowa (2000). This makes it possible to potentially draw from existing algorithms that have already been developed in the scope of each of these frameworks.
- It covers any formal logic, even multiple logics, supporting thus the integration and processing of concepts under various forms of syntactic and semantic heterogeneity. This is important, since we cannot assume conceptual spaces to be represented in a homogeneous manner across diverse domains. Current tools for heterogeneous specifications such as HETS (Mossakowski et al., 2007) allow parsing, static analysis and proof management incorporating various provers and different specification languages.

In this chapter we take the approach of generalising the original notion of amalgam from CBR to be used in the development of a theory of conceptual blending that is close to, and even compatible with, Goguen's work on blending. This means taking a category-theoretic approach to model amalgams in the framework of conceptual blending.

By developing a formal, amalgam-based model of conceptual blending building on Goguen's initial account, we aim at providing general principles that will guide the design of computer systems capable of inventing new higher-level, more abstract concepts and representations out of existing, more concrete concepts and interactions with the environment, and to do so based on the sound reuse and exploitation of existing computational implementations of closely related models such as those for analogical and metaphorical reasoning (Falkenhainer et al., 1989), semantic integration (Schorlemmer and Kalfoglou, 2008), or cognitive coherence (Thagard, 2000). With such a formal, but computationally feasible model we shall ultimately bridge the existing gap between the theoretical foundations of conceptual blending and their computational realisations.

Category theory, although initially designed to describe mathematical entities, has proven to be a successful cornerstone in many computer science applications; a trend which has attracted a lot of attention and researchers, and which has been nicely advocated in Goguen's manifesto paper (Goguen, 1991). One of the most interesting advantages of categorical approaches to computational theories is precisely the fact of being independent of any particular implementation. For this very reason, it is very appealing to search for a categorical framework where a computational theory of conceptual blending based on Fauconnier and Turner's ideas can be developed. In particular, Goguen developed his category-theoretic approach to blending based on colimits, following this basic insight:

> Given a species of structure, say widgets, then the result of interconnecting a system of widgets to form a super-widget corresponds to taking the *colimit* of the diagram of widgets in which the morphisms show how they are interconnected. (Goguen, 1991, Section 6)

In this chapter—after first providing some category theory preliminaries—we shall revisit Goguen's approach that models conceptual blending by means of a certain kind of colimit in ordered categories. Then we present our alternative proposal to model conceptual blending as *amalgams* and conclude the chapter by relating it to the notion of colimit, thus providing a category-theoretic characterisation of amalgams that is computationally realisable.

1.2 Category Theory Preliminaries

In this section, no attempt of being completely self-contained is made, so we suggest the reader supplement the information here provided, whenever necessary, with any standard category theory textbook (e.g., (Barr and Wells, 1990; Pierce, 1991; McLarty, 1992; Mac Lane, 1998)) or short introductions to the subject (e.g., (Diaconescu, 2008, Chapter 2) and (Sannella and Tarlecki, 2012, Chapter 3)).

1.2.1 Categories and Morphisms

Definition 1.1 (Category). A *category* $\mathbf{C}$ consists of the following items:

- A collection $\mathrm{obj}(\mathbf{C})$ of *objects*.
- A collection $\mathrm{hom}(\mathbf{C})$ of *morphisms* (sometimes also called homomorphisms, arrows or maps) satisfying that each morphism f has associated a *source* object denoted by $\mathrm{src}(f)$, and a *target* object denoted by $\mathrm{tg}(f)$. The expression $f\colon A \to B$ is used as a shorthand for claiming that f is a morphism with source A and target B. The collection of all such morphisms is denoted by either $\mathbf{C}(A,B)$ or $\mathrm{hom}(A,B)$.
- For all objects A,B,C, there is a binary associative operation called *composition* from $\mathrm{hom}(A,B) \times \mathrm{hom}(B,C)$ into $\mathrm{hom}(A,C)$. Composition of two morphisms f,g is denoted by writing either

  $$f;g \quad \text{(diagrammatic notation)} \qquad \text{or} \qquad g \circ f \quad \text{(functional notation)}$$

  to refer to the composition of morphisms $f\colon A \to B$ and $g\colon B \to C$.
- For every object A, there is an *identity* morphism id_A belonging to $\mathrm{hom}(A,A)$ which is a neutral element of composition. This neutrality means that
  - $\mathrm{id}_A;f = f$ (for every morphism f with source A)
  - $f;\mathrm{id}_A = f$ (for every morphism f with target A).

Concerning notation to be used later, we point out that $\mathrm{hom}(A,\text{-})$ will denote the collection of all morphisms with source A and $\mathrm{hom}(\text{-},A)$ will denote the collection of morphisms with target A.

Example 1.1 (The categories **Set** *and* **Pfn***).* Among the plethora of examples, there are two well-known categories that are relevant for this chapter (see, e.g., (Calugareanu and Purdea, 2011)).

- The category **Set** has sets as objects and (total) functions as morphisms (endowed with the usual composition of functions).[1]
- The category **Pfn** has sets as objects and partial functions as morphisms (endowed also with the usual composition of functions).

Let us point out that if A and B are finite sets with cardinality n and m, respectively, then $\mathbf{Set}(A,B)$ has cardinality m^n while $\mathbf{Pfn}(A,B)$ has cardinality $(m+1)^n$. In case we have a partial function f, we will use the notation $\mathrm{Dom}(f)$ to refer to its set-theoretical domain and $\mathrm{Im}(f)$ for its set-theoretical image.

Besides using the previously introduced notation $f\colon A \to B$ to refer to morphisms, it is common to use different kinds of graphical arrows to emphasise whether the arrow satisfies some particular property. Thus, we will use

[1] It is worth noticing that, by definition of a category, the collections $\mathrm{hom}(A_1,B_1)$ and $\mathrm{hom}(A_2,B_2)$ must be disjoint unless both $A_1 = A_2$ and $B_1 = B_2$ hold. Thus, for technicality issues it is better to think that a morphism in **Set** is given by an ordered triple (A,f,B) where f is a function from A to B.

- $f\colon A \twoheadrightarrow B$ for *epimorphisms* (i.e., for every $h_1, h_2 \in \hom(B,\text{-})$, if $f;h_1 = f;h_2$ then $h_1 = h_2$).
- $f\colon A \rightarrowtail B$ for *monomorphisms*, (i.e., for every $h_1, h_2 \in \hom(\text{-},A)$, if $h_1;f = h_2;f$ then $h_1 = h_2$).
- $f\colon A \hookrightarrow B$ only for some very special monomorphisms, i.e., those that live in a category whose morphisms are (set-theoretic) functions preserving some structure and which correspond to inclusions.
- $f\colon A \xrightarrow{\simeq} B$ for *isomorphisms* (i.e., there exists some $h \in \hom(B,A)$ such that $f;h = \mathrm{id}_A$ and $h;f = \mathrm{id}_B$).

In the particular cases of **Set** and **Pfn** it is well-known that epimorphisms correspond to being exhaustive on the target object, monomorphisms to injectivity and isomorphisms to bijectivity. Thus, two sets are isomorphic iff they have the same cardinality.

1.2.2 Diagrams, Cocones, and Colimits

Colimits (and also limits) in a category **C** are introduced via diagrams. A *diagram* $\mathscr{D}$ is a functor from a category **J** to the category **C**, and in such a case it is said that $\mathscr{D}$ is **J**-*shaped*. In other words, a *diagram* $\mathscr{D}$ *in* **C** consists of

- a directed graph (where nodes are objects and edges are morphisms in **J**),[2]
- a family (indexed by the set Nodes of nodes of the graph) of objects in **C**, i.e., every node $X \in \mathsf{Nodes}$ of the graph is associated with an object in **C**,
- a family (indexed by the set Edges of edges of the graph) of morphisms in **C** satisfying that: for an edge $f \in \mathsf{Edges}$ between nodes X and Y, the associated morphism has the object associated with X as source, and the object associated with Y as target.

We are mostly interested in the case of *finite diagrams*, i.e., when **J** has a finite number of objects and morphisms. In most such examples, instead of defining the category **J** in words, we will simply draw a directed graph.

Before introducing colimits of a diagram $\mathscr{D}$ in a category **C** we introduce cocones.

Definition 1.2 (Cocone). A *cocone* $\mathfrak{c}$ over a diagram $\mathscr{D}$ in a category **C** is an object O in **C** together with a family (indexed by the nodes in the graph associated with $\mathscr{D}$) $\{\mathfrak{c}_X\}_{X\in\mathsf{Nodes}}$ of morphisms in **C** such that:

- $\mathfrak{c}_X$ has source $\mathscr{D}(X)$, for every node X;
- $\mathfrak{c}_X$ has target O, for every node X;
- $\mathscr{D}(f);\mathfrak{c}_Y = \mathfrak{c}_X$, for every edge f from node X to node Y.

[2] Strictly speaking **J** is the free category generated over the directed graph, but for the purpose of this chapter it is not necessary to worry about this detail.

We refer to the pointed object O, which is called the *apex of* $\mathfrak{c}$, as $\mathrm{apex}(\mathfrak{c})$. The collection of all cocones over $\mathscr{D}$ is denoted by $\mathsf{Cocones}(\mathscr{D},\text{-})$.

Notice that the third condition in Definition 1.2 is expressing a family (one for every edge) of commutativity conditions for triangular graphs; this fact is sometimes emphasised using the terminology *commutative cocone* instead of just saying cocone.

It is rather trivial noticing that every cocone $\mathfrak{c}$ over a diagram $\mathscr{D}$ induces a function $\mathsf{H}_{\mathfrak{c}}$ defined by

$$\begin{array}{rccc} \mathsf{H}_{\mathfrak{c}}\ : & \mathrm{hom}(\mathrm{apex}(\mathfrak{c}),\text{-}) & \longrightarrow & \mathsf{Cocones}(\mathscr{D},\text{-}) \\ & h & \longmapsto & \mathfrak{c};h \end{array}$$

With the notation $\mathfrak{c};h$ we obviously refer to the family $\{g;h\}_{g \text{ is a morphism in } \mathfrak{c}}$, i.e., $\{\mathfrak{c}_X;h\}_{X\in\mathsf{Nodes}}$. These induced functions can be used to define that two cocones $\mathfrak{c}$ and $\mathfrak{d}$ (over the same diagram) are *isomorphic* when there is some isomorphism h in $\mathbf{C}$ such that $\mathfrak{d} = \mathsf{H}_{\mathfrak{c}}(h)$.

Definition 1.3 (Colimit). A cocone $\mathfrak{c}$ over a diagram $\mathscr{D}$ in a category $\mathbf{C}$ is said to be a *colimit* if the function $\mathsf{H}_{\mathfrak{c}}$ is a bijection. We write $\mathfrak{colim}(\mathscr{D},\mathbf{C})$, or simply $\mathfrak{colim}(\mathscr{D})$, to refer to a colimit; and we will use $\mathsf{colim}(\mathscr{D},\mathbf{C})$ or $\mathsf{colim}(\mathscr{D})$ for the apex in the cocone $\mathfrak{colim}(\mathscr{D},\mathbf{C})$.

It is worth noticing that Definition 1.3 can be rephrased as claiming that every cocone over $\mathscr{D}$ is of the form $\mathfrak{c};h$ for some unique morphism h. This remark allows us to rewrite the existence of a colimit as saying that: for every cocone over the same diagram, there is exactly one solution for a univariate system, using the cocone as parameters, of morphism equations. As an example, we illustrate this fact for the case of a colimit of a span (a V-shaped diagram), which is also called *pushout*.

Definition 1.4 (Pushout). Given a diagram $B \xleftarrow{f} A \xrightarrow{g} C$ —called *span* or V-shaped diagram—a *pushout* of this span is a colimit (see Definition 1.3), i.e., it is a cocone $\begin{array}{ccccc} & & \mathrm{apex}(\mathfrak{c}) & & \\ & \overset{\mathfrak{c}_B}{\nearrow} & \uparrow & \overset{\mathfrak{c}_C}{\nwarrow} & \\ B & & \mathfrak{c}_A & & C \\ & & A & & \end{array}$ such that whenever $\begin{array}{ccccc} & & D & & \\ & \overset{\mathfrak{d}_B}{\nearrow} & \uparrow & \overset{\mathfrak{d}_C}{\nwarrow} & \\ B & & \mathfrak{d}_A & & C \\ & \underset{f}{\nwarrow} & | & \underset{g}{\nearrow} & \\ & & A & & \end{array}$ commutes, it holds that the univariate system

$$\mathfrak{c}_B;h = \mathfrak{d}_B \qquad \mathfrak{c}_A;h = \mathfrak{d}_A \qquad \mathfrak{c}_C;h = \mathfrak{d}_C$$

of morphism equations has a unique solution for h.

For each categorical construct such as cocones, colimits, pushouts and spans, there exists also a dual notion with morphsims pointing in the opposite direction, such as cones, limits, pullbacks and cospans. We refer the reader to the literature for a thorough discussion of these (e.g., (Pierce, 1991)).

It is always the case that two colimits over the same diagram are isomorphic cocones, i.e., *colimits are unique up to isomorphism*. Indeed, if $\mathfrak{c}$ is a colimit, then the collection of all colimits is exactly $\{\mathfrak{c};h \mid h$ is an ismorphism with $\mathrm{src}(h) = \mathrm{apex}(\mathfrak{c})\}$. On the other hand, the existence of a colimit is, in general, not guaranteed; it depends very much on the diagram $\mathscr{D}$ and the category $\mathbf{C}$.

Let us now mention two facts that restrict which cocones can be a colimit. The first fact is a trivial consequence of the injectivity of the function $\mathsf{H}_{\mathfrak{c}}$: all colimits $\mathfrak{c}$ have to be *jointly epimorphic*, which means that whenever h_1 and h_2 are two morphisms with source $\mathrm{apex}(\mathfrak{c})$ and such that "$\mathfrak{c}_X;h_1 = \mathfrak{c}_X;h_2$ for every node X", then $h_1 = h_2$.[3] The second fact, also obvious from Definition 1.3, is that for every object E, the set $\mathsf{Cocones}(\mathscr{D},E)$ (i.e., the collection of cocones over $\mathscr{D}$ with apex E) must have the same cardinality as the set $\hom(\mathrm{apex}(\mathfrak{c}),E)$. These two facts are, in general, very powerful tools to recognise possible candidates as a colimit over a diagram. In the particular cases of **Set** and **Pfn** the second fact can be used to completely determine the possible apexes of colimits (since all objects with the same cardinal are isomorphic). Remark 1.1 describes the method for the case of **Pfn**.

Remark 1.1 (Cardinality trick for **Pfn***).* Consider the natural number m of cocones over $\mathscr{D}$ with apex $\{①\}$ (i.e., a singleton set). Then, the cardinal of an object $\mathsf{colim}(\mathscr{D},\mathbf{Pfn})$ has to be the only natural number n such that $m = 2^n$.

Definition 1.5. A category is said to be *cocomplete* in case that for all diagrams in $\mathbf{C}$ there is a colimit. Analogously, *complete* refers to the existence of all limits; and *bicomplete* refers to being both complete and cocomplete.

The categories **Set** and **Pfn** introduced in Example 1.1 are well-known to be bicomplete. Moreover, it is also known that if all morphisms of a diagram $\mathscr{D}$ in **Pfn** are total functions (i.e., the diagram lives inside **Set**) then $\mathfrak{colim}(\mathscr{D},\mathbf{Set}) = \mathfrak{colim}(\mathscr{D},\mathbf{Pfn})$, i.e., it does not matter whether one computes the colimit in **Set** or in **Pfn**. Let us mention that this last remark is known to be false for the case of limits.[4]

[3] Is is worth pointing out that when $\mathbf{C}$ has coproducts, the following (i) and (ii) are equivalent. (i) $\{\mathfrak{c}_X\}_{X\in\mathsf{Nodes}}$ is jointly epimorphic; (ii) the single morphism $\bigoplus\{\mathfrak{c}_X\}_{X\in\mathsf{Nodes}}$ is epimorphic. This relationship explains the intuition behind this "jointly" terminology.

[4] An easy counterexample can be obtained considering the categorical product of two singleton sets, for example, $A := \{$✈$\}$ and $B := \{$➽$\}$. A quick way to convince oneself that the categorical product computed in **Set** is different than in **Pfn** is to use the cardinality trick described in Remark 1.1 (but dualised, in order to use it for limits instead of colimits). The fact that there are exactly four cones in **Pfn** with apex $\{①\}$ (i.e., a singleton) forces that the product in **Pfn** must have three elements; on the other hand, using that there is exactly one cone in **Set** with apex $\{①,②\}$ one deduces that the product in **Set** must have one element.

Indeed, the content of the previous paragraph is generalised in the following well-known statement (see (Poigné, 1986, p. 20)):

- the product in **Set** of A and B is given by the cone $A \xleftarrow{\pi_A} O \xrightarrow{\pi_B} B$ where O is the Cartesian product of A and B (i.e., $O := A \times B$), and the morphisms π_A and π_B are the "projections" from the Cartesian product.

1.2.3 Partial Morphisms

To finish this section about category theory preliminaries we introduce a category that will play a role in Section 1.5, where we discuss the relationship between colimits and amalgams, and their role in modelling conceptual blending. Our aim with this category is to capture the notion of *partial morphism*, which models the selective projection, in conceptual blending, of parts of the input spaces into the blend space.

Definition 1.6. Let $\mathbf{C}$ be a category that is closed under pullbacks, i.e., the limits of all cospans exist. The category $\mathbf{Pfn}(\mathbf{C})$ has the same objects as $\mathbf{C}$, and a morphism from an object A to an object B is the isomorphism class[5] of the *mono spans* from A to B, which are defined to be the spans $A \overset{f}{\leftarrowtail} D \overset{g}{\longrightarrow} B$ where f is a monomorphism in $\mathbf{C}$. Composition of spans $A \overset{f}{\longleftarrow} D \overset{g}{\longrightarrow} B$ and $B \overset{h}{\longleftarrow} E \overset{l}{\longrightarrow} C$ is defined (up to isomorphism) using the cone

$$\begin{array}{ccccc} D & & B & & E \\ & \nwarrow^{\mathfrak{c}_D} & \uparrow{\scriptstyle \mathfrak{c}_B} & \nearrow_{\mathfrak{c}_E} & \\ & & \mathrm{apex}(\mathfrak{c}) & & \end{array}$$

obtained as the pullback of $D \overset{g}{\longrightarrow} B \overset{h}{\longleftarrow} E$. The result of the composition is by definition the span $A \overset{\mathfrak{c}_D;f}{\longleftarrow} \mathrm{apex}(\mathfrak{c}) \overset{\mathfrak{c}_E;l}{\longrightarrow} C$. A *partial morphism* from A to B is defined as the isomorphism class of a mono span $A \overset{f}{\leftarrowtail} D \overset{g}{\longrightarrow} B$. Thus, the morphisms in $\mathbf{Pfn}(\mathbf{C})$ are nothing else than the partial morphisms.

It is well known that $\mathbf{Pfn}(\mathbf{Set})$ is (categorically) equivalent to the category $\mathbf{Pfn}$ (and also equivalent to the category of pointed sets). Even more, $\mathbf{Pfn}(\mathbf{Set})$ and $\mathbf{Pfn}$ are isomorphic categories: there is an obvious bijection between partial morphisms in $\mathbf{Set}$ and morphisms in $\mathbf{Pfn}$. Thus, $\mathbf{Pfn}(\mathbf{C})$ can be considered as a natural candidate for generalising the category $\mathbf{Pfn}$ of partial functions.

- the product in **Pfn** of A and B is given by the cone $\begin{array}{ccc} A & & B \\ {}^{\mathfrak{c}_A}\nwarrow & & \nearrow^{\mathfrak{c}_B} \\ & O & \end{array}$ where $O := (A \times B) \oplus A \oplus B$ (here $\oplus$ refers, as above, to the disjoint union), the morphism $\mathfrak{c}_A$ is $\pi_A \oplus \mathrm{id}_A \oplus \emptyset$, and the morphism $\mathfrak{c}_B$ is $\pi_B \oplus \emptyset \oplus \mathrm{id}_B$.

The last statement is providing the intuition that for the product in **Pfn** of two sets one needs to consider the ordered pairs in the Cartesian product, but also add those ordered pairs that are missing one element of the pair.

[5] In other words, the spans $A \overset{f}{\longleftarrow} D \overset{g}{\longrightarrow} B$ and $A \overset{f'}{\longleftarrow} D' \overset{g'}{\longrightarrow} B$ are considered equal when there is an isomorphism $h : D \to D'$ such that $\begin{array}{ccccc} & & D & & \\ & {}^{f}\swarrow & \downarrow{\scriptstyle h} & \searrow^{g} & \\ A & & & & B \\ & {}_{f'}\nwarrow & & \nearrow_{g'} & \\ & & D' & & \end{array}$ commutes.

Among partial morphisms from A to B there are some outstanding ones which we call *total*. They are, by definition, the isomorphism classes of mono spans $A \overset{f}{\leftarrowtail} D \overset{g}{\longrightarrow} B$ where f is an isomorphism. It is obvious that the total morphisms form a subcategory (i.e., total morphisms are closed under composition and the identities are total) of $\mathbf{Pfn}(\mathbf{C})$, and this subcategory is equivalent to $\mathbf{C}$.

The categories $\mathbf{Pfn}(\mathbf{C})$ of partial morphisms are well known in the literature. They were first considered in (Robinson and Rosolini, 1988) within an even more general setting; there the authors introduce for every class $\mathscr{M}$ of monomorphisms satisfying certain constraints (see (Hayman and Heindel, 2014, Definitions 6 and 7) for a modern presentation) a category $\mathbf{Pfn}(\mathbf{C},\mathscr{M})$. Our category $\mathbf{Pfn}(\mathbf{C})$ corresponds to choosing $\mathscr{M}$ as the class of all monomorphisms. As for now, we have decided to avoid this more general framework for the sake of simplicity.

1.3 Conceptual Blending as Colimits

The aim of this section is to explain Goguen's framework for conceptual blending. This framework is developed in (Goguen, 1999) (mainly in Section 5 and Appendix B), and instead of using plain categories it is based on categories enriched with a partial order on morphisms.

(Kutz et al., 2012) and (Kutz et al., 2014) use Goguen's categorical framework, but without ordered categories, i.e., only plain categories are considered. The proposed framework uses the category of CASL theories, which is known to be cocomplete (Mossakowski, 1998), and whose computation of colimits is supported in HETS.[6] Besides this, the authors of (Kutz et al., 2012, 2014) also advocate for using the distributed ontology language DOL as a metalanguage for specifying categorical diagrams (i.e., families of morphisms). When computing colimits, they point out (indeed Goguen already did) that in some case it might be interesting (for blending purposes) to ignore some of the morphisms in the diagram, and consider them just as auxiliary morphisms.

An important difference between (Kutz et al., 2012) and (Kutz et al., 2014) is that in (Kutz et al., 2014) the authors only focus on input diagrams given by total functions, while in the previous version (Kutz et al., 2012) the same authors consider a more general setting allowing for partial morphisms. This simplification has deep consequences, because the colimits of diagrams formed by total functions are, in most cases, although computed in categories of partial morphisms, formed only by total functions (see Page 10).

[6] Colimits are available in HETS without problems in the homogeneous case of reasonable institutions (which include most cases: first-order logic, description logics, etc.), but things are not so simple in the heterogeneous case; for such a case only the colimits of certain diagrams (the 'connected thin inf-bounded' ones) (Codescu and Mossakowski, 2008) are computed.

1.3.1 Ordered Categories

Definition 1.7 (Ordered category). An *ordered category* is a category $\mathbf{C}$ such that

- for every two objects A and B, there is a partial order $\sqsubseteq_{A,B}$ on the set $\hom(A,B)$;
- composition is monotonic with respect to $\sqsubseteq$ in both arguments (i.e., if $f_1 \sqsubseteq g_1$ and $f_2 \sqsubseteq g_2$, then $f_1;f_2 \sqsubseteq g_1;g_2$).

Concerning notation, it is customary to omit indices and simply use $\sqsubseteq$ (see second item), i.e., $\sqsubseteq$ can be considered to be $\bigcup\{\sqsubseteq_{A,B} | \, A,B \in \mathrm{obj}(\mathbf{C})\}$.

Ordered categories are a special case of so-called 2-categories (see (Leinster, 2002; Johnstone, 2002; Lack, 2010)). Here, there is at most one 2-cell between two 1-cells (i.e., morphisms). Thus, ordered categories lie between plain 1-categories and 2-categories. For this reason, Goguen (1999) introduces the term $\frac{3}{2}$*-categories* to refer to ordered categories.[7] Other names have also been used in the literature, such as locally partially ordered categories, locally posetal categories, Pos-enriched categories, order-enriched categories, etc. We refer to (Kahl, 2010) for a detailed approach to ordered categories, without considering all the difficulties that arise when dealing with general 2-categories.

Example 1.2. The categories $\mathbf{Pfn}(\mathbf{C})$ are ordered categories in the following sense: consider two partial morphisms from A to B, given respectively by the isomorphism classes of the mono spans

$$A \overset{f}{\leftarrowtail} D \overset{g}{\longrightarrow} B \qquad \text{and} \qquad A \overset{f'}{\leftarrowtail} D' \overset{g'}{\longrightarrow} B \;\; .$$

We say that the first partial morphism is *below* the second one (denoted $\sqsubseteq$) if there is a morphism $h : D \to D'$ such that

$$\begin{array}{ccccc} & & D & & \\ & {\scriptstyle f}\swarrow & & \searrow{\scriptstyle g} & \\ A & & \Big\downarrow{\scriptstyle h} & & B \\ & {\scriptstyle f'}\nwarrow & & \nearrow{\scriptstyle g'} & \\ & & D' & & \end{array}$$

commutes. In such a case, h is also a monomorphism, and $\sqsubseteq$ is a partial order: antisymmetry is obtained using the cancellativity property given by monomorphisms. Moreover, the partial morphisms that are total are the maximal elements of the partial order $\sqsubseteq$ just defined. We will refer to this partial order $\sqsubseteq$ as the *extension partial order*.

Example 1.2 tells us, in particular, that $\mathbf{Pfn}(\mathbf{Set})$ is an ordered category; for this case it holds that

[7] The definition given in (Goguen, 1999, Definition 6) also states that the identity morphism id_A has to be maximal in $\hom(A,A)$. We do not require this last condition in the definition we ultimately decided to adopt, but this property also holds for the most natural examples of ordered categories (see Example 1.2).

$$f \sqsubseteq g \qquad \text{iff} \qquad \text{whenever } f \text{ is defined, } g \text{ is also defined and it agrees with } f.$$

Moreover, the structure of the partial order $\sqsubseteq$ resembles (but is not) a lattice because:

- for every two partial morphisms f_1 and f_2 (with the same sources and targets), there is also a partial morphism $f_1 \sqcap f_2$ which is the infimum in $\sqsubseteq$;
- for every two partial morphisms f_1 and f_2, if they are *compatible* (i.e., if there is some g such that $f_1 \sqsubseteq g$ and $f_2 \sqsubseteq g$) then there is also a partial morphism $f_1 \sqcup f_2$ which is the supremum in $\sqsubseteq$.

It is also worth noticing that the partial orders $\sqsubseteq_{A,B}$ are *directed-complete partial orders* (*dcpo*), which means that every directed subset has a supremum (which we will denote using the symbol $\bigsqcup$). And the composition function can be checked to be *Scott-continuous*, which means that, for every directed family $\{g_i \mid i \in I\}$ of partial functions and every partial function f,

- $\{f;g_i \mid i \in I\}$ is also directed and its supremum is $f;\bigsqcup\{g_i \mid i \in I\}$;
- $\{g_i;f \mid i \in I\}$ is also directed and its supremum is $\bigsqcup\{g_i \mid i \in I\};f$.

Notice also that **Set** is equivalent to the subcategory of **Pfn**(**Set**) given by total morphisms.

1.3.2 Colimits in Ordered Categories

In the context of ordered categories there are, at least, two very natural alternative possibilities concerning colimits (see (Kahl, 2010, Chapter 4)). One of them produces a strengthening of the plain notion of colimit, and we will refer to them as *ordered colimits*. The other one accepts a more general class of diagrams, which instead of considering functors considers so-called *lax functors*, where commutativity is replaced with semicommutativity. The latter follows a very similar pattern than the one given for colimits in Definition 1.3, and the respective colimits are called *lax colimits*.

Definition 1.8 (Ordered colimit, see (Kahl, 2010, Definition 4.1.2)). A cocone $\mathfrak{c}$ over a diagram $\mathscr{D}$ in an ordered category $\mathbf{C}$ is said to be an *ordered colimit* in case that the function $\mathsf{H}_{\mathfrak{c}}$ introduced on Page 9 is an order-isomorphism (and therefore also a bijection) between the partial orders $\langle \mathrm{hom}(\mathrm{apex}(\mathfrak{c}),\text{-}), \sqsubseteq \rangle$ and $\langle \mathsf{Cocones}(\mathscr{D},\text{-}), \sqsubseteq^* \rangle$. The order $\sqsubseteq^*$ considered among cocones is the one defined component-wise, that is, given two cocones $\mathfrak{c} := \{\mathfrak{c}_X\}_{X \in \mathsf{Nodes}}$ and $\mathfrak{d} := \{\mathfrak{d}_X\}_{X \in \mathsf{Nodes}}$ with the same apex, it holds that

$$\mathfrak{c} \sqsubseteq^* \mathfrak{d} \qquad \text{iff} \qquad \mathfrak{c}_X \sqsubseteq \mathfrak{d}_X \text{ for every node } X \in \mathsf{Nodes}.$$

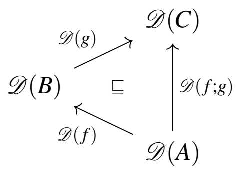
From Definition 1.8 it is obvious that, if $\mathfrak{c}$ is an ordered colimit, then: whenever h_1 and h_2 are two morphisms with source $\mathrm{apex}(\mathfrak{c})$ and '$\mathfrak{c}_X; h_1 \sqsubseteq \mathfrak{c}_X; h_2$ for every node X', then $h_1 \sqsubseteq h_2$. We will refer to such condition as being *jointly semiepimorphic*.[8]

In the particular case of the ordered category **Pfn** (with the extension partial order described in Example 1.2), one can check that colimits are also ordered colimits.

Next, in order to introduce lax colimits we need to firstly introduce lax diagrams and lax cocones. The only difference between a functor $\mathscr{D} : \mathbf{J} \longrightarrow \mathbf{C}$ and a lax functor $\mathscr{D} : \mathbf{J} \longrightarrow \mathbf{C}$ is that instead of equality one only requires

$$\mathrm{id}_{\mathscr{D}(A)} \sqsubseteq \mathscr{D}(\mathrm{id}_A) \qquad \text{and} \qquad \mathscr{D}(f);\mathscr{D}(g) \sqsubseteq \mathscr{D}(f;g).$$

The second condition is known as *semicommutativity*, and it is common to represent it graphically as follows:



Notice that if the ordered category satisfies that the identity morphisms are maximal, then the first condition $\mathrm{id}_{\mathscr{D}(A)} \sqsubseteq \mathscr{D}(\mathrm{id}_A)$ can be rewritten as saying $\mathrm{id}_{\mathscr{D}(A)} = \mathscr{D}(\mathrm{id}_A)$. A *lax diagram* in an ordered category $\mathbf{C}$ is defined to be a lax functor $\mathscr{D} : \mathbf{J} \longrightarrow \mathbf{C}$. Here $\mathbf{J}$ is just a category (not necessarily an ordered category).

A *lax cocone* $\mathfrak{c}$ over a lax diagram $\mathscr{D}$ in a category $\mathbf{C}$ is an object O in $\mathbf{C}$ together with a family (indexed by the nodes in the graph associated with $\mathscr{D}$) $\{\mathfrak{c}_X\}_{X \in \mathsf{Nodes}}$ of morphisms in $\mathbf{C}$ such that:

- $\mathfrak{c}_X$ has source $\mathscr{D}(X)$, for every node X;
- $\mathfrak{c}_X$ has target O, for every node X;
- $\mathscr{D}(f); \mathfrak{c}_Y \sqsubseteq \mathfrak{c}_X$ for every edge f from node X to node Y.

Thus, lax cocones are capturing the intuition of *semicommutative cocones*. As expected we will refer to the apex object as $\mathrm{apex}(\mathfrak{c})$. The collection of all lax cocones over $\mathscr{D}$ will be denoted by $\mathsf{laxCocones}(\mathscr{D}, \text{-})$.

It is rather trivial noticing that every lax cocone $\mathfrak{c}$ over a lax diagram $\mathscr{D}$ induces a function[9] $\mathsf{H}_{\mathfrak{c}}$ defined by

$$\begin{array}{rccc} \mathsf{H}_{\mathfrak{c}} : & \mathrm{hom}(\mathrm{apex}(\mathfrak{c}), \text{-}) & \longrightarrow & \mathsf{laxCocones}(\mathscr{D}, \text{-}) \\ & h & \longmapsto & \mathfrak{c}; h \end{array}$$

[8] When there are ordered coproducts (in the sense of Definition 1.8) it is obvious that this definition also follows the same intuition explained in Section 1.2.2. That is, $\{\mathfrak{c}_X\}_{X \in \mathsf{Nodes}}$ is jointly semiepimorphic iff the single morphism $\bigoplus \{\mathfrak{c}_X\}_{X \in \mathsf{Nodes}}$ is so.

[9] We use the same notation $\mathsf{H}_{\mathfrak{c}}$ as for the case of plain categories and colimits, but this is not a problem because the context always clarifies which one we refer to.

Definition 1.9 (Lax colimit, see (Kahl, 2010, Definition 4.3.2)). A lax cocone $\mathfrak{c}$ over a lax diagram $\mathscr{D}$ in an ordered category $\mathbf{C}$ is said to be a *lax colimit* when the recently introduced function $\mathsf{H}_{\mathfrak{c}}$ is an order-isomorphism (and hence a bijection) between the partial orders $\langle \mathsf{hom}(\mathsf{apex}(\mathfrak{c}), \text{-}), \sqsubseteq \rangle$ and $\langle \mathsf{laxCocones}(\mathscr{D}, \text{-})), \sqsubseteq^* \rangle$. The ordered $\sqsubseteq^*$ considered among lax cocones is the one defined component-wise (see Definition 1.8).

It is again obvious that lax colimits must be jointly semiepimorphic. Notice also that in case of considering a diagram $\mathscr{D}$ (instead of an arbitrary lax diagram), the notions of lax colimit and ordered colimit collapse (up to isomorphism) if and only if all lax cocones are cocones. Thus, whenever semicommutativity is not trivially reduced to commutativity, the two recently introduced notions of colimits can be different.

1.3.3 $\frac{3}{2}$-Colimits

It is well-known that the cocone of an ordered colimit is unique up to isomorphism. And the same happens for the lax cocone of a lax colimit. Goguen considers these facts to show that they might not be adequate notions for the formalisation of conceptual blending, since one expects more than one way to blend concepts. For this reason he proposes the following alternative notion.[10]

Definition 1.10 ($\frac{3}{2}$-Colimit, see (Goguen, 1999, Definition 12)). A lax cocone $\mathfrak{c}$ over a lax diagram $\mathscr{D}$ in an ordered category $\mathbf{C}$ is said to be a $\frac{3}{2}$*-colimit* in case that, for every lax cocone $\mathfrak{d}$ (with apex D) over $\mathscr{D}$, it holds that the set

$$\{h \mid \mathsf{H}_{\mathfrak{c}}(h) \sqsubseteq^* \mathfrak{d}\} \qquad \text{(which is a subset of } \mathsf{hom}(\mathsf{apex}(\mathfrak{c}), D))$$

has a maximum element on $\sqsubseteq$.

Notice that this last definition is equivalent to just saying that the function

$$\begin{array}{rccc} \mathsf{H}_{\mathfrak{c}} : & \langle \mathsf{hom}(\mathsf{apex}(\mathfrak{c}), \text{-}), \sqsubseteq \rangle & \longrightarrow & \langle \mathsf{laxCocones}(\mathscr{D}, \text{-}), \sqsubseteq^* \rangle \\ & h & \longmapsto & \mathfrak{c}; h \end{array}$$

fulfills that the anti-image of principal downsets (i.e., downsets of an element) are also principal downsets.[11] This last restatement of the notion of $\frac{3}{2}$-colimits has the advantage of providing an easier comparison with Definition 1.9. In particular, it becomes obvious that if $\mathfrak{c}$ is a lax colimit over $\mathscr{D}$, then it is also a $\frac{3}{2}$-colimit.

When the ordered category involves partial orders that are dcpos and composition is Scott-continuous, then it is worth noticing that the following statements are equivalent:[12]

[10] In (Goguen, 2001, Section 3.1) the expression "lax pushouts" is used in a naive way: this has not to be understood as a particular case of lax colimits in ordered categories.

[11] The downset of an element h is the set of all $g \sqsubseteq h$.

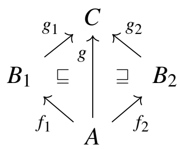
[12] The assumptions just stated are only necessary to prove the implication $2 \Rightarrow 1$; the reverse implication always holds.

1. The set $\{h \mid \mathsf{H}_{\mathfrak{c}}(h) \sqsubseteq^* \mathfrak{d}\}$ has a maximum element on $\sqsubseteq$.
2. The set $\{f \mid \mathsf{H}_{\mathfrak{c}}(h) \sqsubseteq^* \mathfrak{d}\}$ is directed, i.e., whenever $\mathsf{H}_{\mathfrak{c}}(h_1) \sqsubseteq^* \mathfrak{d}$ and $\mathsf{H}_{\mathfrak{c}}(h_2) \sqsubseteq^* \mathfrak{d}$ then there is some g such that $h_1 \sqsubseteq g$, $h_2 \sqsubseteq g$ and $\mathsf{H}_{\mathfrak{c}}(g) \sqsubseteq^* \mathfrak{d}$.

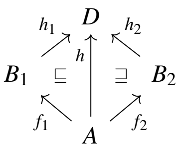
Notice that the first condition is the one involved in Definition 1.10, and also that **Pfn** satisfies the hypotheses for such equivalence.

For the case of the diagram $B_1 \xleftarrow{f_1} A \xrightarrow{f_2} B_2$, Definition 1.10 provides the notion of $\frac{3}{2}$-*pushouts*, which is Goguen's proposal for a formalisation of blending. We restate his proposal in Definition 1.11.

Definition 1.11 ($\frac{3}{2}$-Pushout). A $\frac{3}{2}$-pushout of a span $B_1 \xleftarrow{f_1} A \xrightarrow{f_2} B_2$ is given by a lax cocone



satisfying that whenever

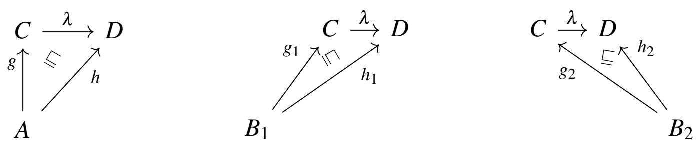


semicommutes, it holds that the univariate system

$$g;\lambda \sqsubseteq h \qquad g_1;\lambda \sqsubseteq h_1 \qquad g_2;\lambda \sqsubseteq h_2$$

of morphism equations has a maximum solution for the indeterminate λ.

The formulation given in Definition 1.11 for presenting $\frac{3}{2}$-pushouts exhibits an obvious relationship with the one given in Definition 1.4; the main difference is that instead of looking for unique solutions to a family of morphism equations one looks for the best (i.e., largest) solution to a family of morphism inequations. For the particular inequations given in Definition 1.11, the family of morphism inequations is the one stating that the three triangles



semicommute.

It is worth saying that whenever the category **C** has ordered coproducts (in the sense of Definition 1.8) the system $\{\mathfrak{c}_X;h \sqsubseteq \mathfrak{d}_X \mid X \in \mathsf{Node}\}$ of morphism inequations (that is, the one which appears in Definition 1.10) is equivalent to the following single inequation: $(\bigoplus\{\mathfrak{c}_X \mid X \in \mathsf{Node}\});h \sqsubseteq \bigoplus\{\mathfrak{d}_X \mid X \in \mathsf{Node}\}$.

Let us assume now that $\mathfrak{c}$ is a $\frac{3}{2}$-colimit (with apex C) over a lax diagram $\mathscr{D}$ and that $h \in \hom(C,D)$. Then, by monotonicity it holds that $\mathfrak{c};h$ is also a lax cocone (with apex D). Therefore, by definition of $\frac{3}{2}$-colimit the univariate inequational system $\mathfrak{c};\lambda \sqsubseteq^* \mathfrak{c};h$ has a maximum solution for λ. In other words, the inequational system

$$\mathfrak{c}_X;\lambda \sqsubseteq \mathfrak{c}_X;h \qquad \text{for every node } X$$

has a maximum solution for λ. We denote such a maximum solution g. Considering that h is also trivially a solution to the very system, we obtain that $h \sqsubseteq g$. Thus, by monotonicity it must hold that $\mathfrak{c}_X;h \sqsubseteq \mathfrak{c}_X;g$ for every node X. Therefore, g is also the largest solution to the equational system $\mathfrak{c};\lambda = \mathfrak{c};h$.

Thus, we have demonstrated that for every $\frac{3}{2}$-colimit $\mathfrak{c}$ (with object C) over a lax diagram $\mathscr{D}$ and every $h \in \hom(C,\text{-})$, there exists $\max_{\sqsubseteq}\{g \mid \mathsf{H}_{\mathfrak{c}}(g) = \mathsf{H}_{\mathfrak{c}}(h)\}$ that coincides with $\max_{\sqsubseteq}\{g \mid \mathsf{H}_{\mathfrak{c}}(g) \sqsubseteq \mathsf{H}_{\mathfrak{c}}(h)\}$. Thus, for every $\frac{3}{2}$-colimit $\mathfrak{c}$ over a lax diagram $\mathscr{D}$, we can define the *expansion* function

$$\begin{aligned} \mathsf{xpan}_{\mathfrak{c}} : \hom(\mathrm{apex}(\mathfrak{c}),\text{-}) &\longrightarrow \hom(\mathrm{apex}(\mathfrak{c}),\text{-}) \\ h &\longmapsto \mathsf{xpan}_{\mathfrak{c}}(h) := \max_{\sqsubseteq}\{g \mid \mathsf{H}_{\mathfrak{c}}(g) = \mathsf{H}_{\mathfrak{c}}(h)\} = \\ &\qquad \max_{\sqsubseteq}\{g \mid \mathsf{H}_{\mathfrak{c}}(g) \sqsubseteq \mathsf{H}_{\mathfrak{c}}(h)\} \end{aligned}$$

It is obvious that $\mathsf{H}_{\mathfrak{c}}(h) = \mathsf{H}_{\mathfrak{c}}(\mathsf{xpan}_{\mathfrak{c}}(h))$. Moreover, this function $\mathsf{xpan}_{\mathfrak{c}}$ is

- extensive, i.e., $h \sqsubseteq \mathsf{xpan}_{\mathfrak{c}}(h)$;
- increasing, i.e., if $h_1 \sqsubseteq h_2$ then $\mathsf{xpan}_{\mathfrak{c}}(h_1) \sqsubseteq \mathsf{xpan}_{\mathfrak{c}}(h_2)$;
- idempotent, i.e., $\mathsf{xpan}_{\mathfrak{c}}(\mathsf{xpan}_{\mathfrak{c}}(h)) = \mathsf{xpan}_{\mathfrak{c}}(h)$.

Consequently, every $\frac{3}{2}$-colimit $\mathfrak{c}$ induces a closure operator (or closure system) (Burris and Sankappanavar, 2012, Section I.5) on the set $\hom(\mathrm{apex}(\mathfrak{c}),\text{-})$.

On Page 10 we point out that colimits are jointly epimorphic. Unfortunately, in the arbitrary case it not so clear whether this property also holds for $\frac{3}{2}$-colimits. However, as is obvious from the definitions of $\mathsf{xpan}_{\mathfrak{c}}$, it holds that

if h_1 and h_2 satisfy that $\mathsf{H}_{\mathfrak{c}}(h_1) = \mathsf{H}_{\mathfrak{c}}(h_2)$, then $\mathsf{xpan}_{\mathfrak{c}}(h_1) = \mathsf{xpan}_{\mathfrak{c}}(h_2)$.

In other words, the following property (which resembles the definition of jointly epimorphic) holds for $\frac{3}{2}$-colimits $\mathfrak{c}$:

if h_1 and h_2 satisfy that '$\mathfrak{c}_X;h_1 = \mathfrak{c}_X;h_2$ for every node X', then $\mathsf{xpan}_{\mathfrak{c}}(h_1) = \mathsf{xpan}_{\mathfrak{c}}(h_2)$.

It is worth noticing that $\mathsf{xpan}_{\mathfrak{c}}(h_1) = \mathsf{xpan}_{\mathfrak{c}}(h_2)$ implies, in particular, that h_1 and h_2 are compatible.

Goguen's proposal is to use $\frac{3}{2}$-pushouts as a computational method for finding conceptual blends (see Figure 1.1). In the easiest case (i.e., the blend of two concepts), this framework assumes that we have previously chosen

- a morphism f_1 from the generic space G into input space I_1 (i.e., $f_1 : G \to I_1$), and also

- a morphism f_2 from the generic space G into input space I_2 (i.e., $f_2 : G \to I_2$).

Furthermore, Goguen suggests to consider all $\frac{3}{2}$-pushouts of the span $I_1 \stackrel{f_1}{\leftarrow} G \stackrel{f_2}{\rightarrow} I_2$ as candidates for blending of the two initial concepts. In the examples provided in (Goguen, 1999)[13] this is done using ordered categories whose objects are algebraic theories (using the formal specification language OBJ), morphisms correspond to partial functions preserving the structure, and the partial order corresponds to being an extension.

There are several difficulties in order to provide a computational framework to conceptual blending following Goguen's categorical proposal. Some of them are as follows.

- While there are several available software packages for dealing with "algebraic theory" categories and colimits (like HETS (Mossakowski et al., 2007; Codescu et al., 2010)) this is not the case in the context of ordered categories.
- Although (Goguen, 1999) contains a first theoretical study of $\frac{3}{2}$-colimits, the theoretical framework still needs to be improved before considering computational implementations. For example, can we characterise all $\frac{3}{2}$-pushouts in the ordered category **Pfn**? What about more complex diagrams that are still in **Pfn**? What about considering other well-known ordered categories? Can we get rid of the ordered category **C** appealing to some particular plain category built from **C**?

For this reason we propose an alternative proposal to model conceptual blending, basing it on the notion of *amalgam*.

1.4 Conceptual Blending as Amalgams

An amalgam is a description that combines parts of two other descriptions as a new coherent whole. There are notions that are related to amalgams in addition to conceptual blending, notions such as merging operation or information fusion. They all have in common that they deal with combining information from more than one 'source' into a new integrated and coherent whole; their differences reside on the assumptions they make on the sources characteristics' and the way in which the combination of the sources takes place.

The notion of amalgams was developed in the context of Case-Based Reasoning (CBR), where new problems are solved based on previously solved problems or cases, residing on a case base (Ontañón and Plaza, 2010). Solving a new problem often requires more than one case from the case base, so their content has to be combined in some way to solve the new problem. The notion of amalgam of two cases—two descriptions of problems and their solutions, or situations and their

[13] It is also worth looking at `http://cseweb.ucsd.edu/~goguen/papers/blend.html` because this site has more recent examples.

outcomes—is a proposal to formalise this process of the ways in which they can be combined to produce a new, coherent case.

Formally, the notion of amalgams can be defined in any representation language $\mathscr{L}$ for which a subsumption relation $\sqsubseteq$ between the terms (or descriptions) of $\mathscr{L}$ can be defined. We say that a term ψ_1 subsumes another term ψ_2 ($\psi_1 \sqsubseteq \psi_2$) when ψ_1 is more general than (or equal to) ψ_2.[14]

Additionally, we assume that $\mathscr{L}$ contains the infimum element $\perp$ (or 'any') and the supremum element $\top$ (or 'none') with respect to the subsumption order.

Next, for any two terms ψ_1 and ψ_2 we can define their *unification*, ($\psi_1 \sqcup \psi_2$), which is the *most general specialisation* of two given terms, and their *anti-unification*, defined as the *least general generalisation* of two terms, representing the most specific term that subsumes both. Intuitively, a unifier (if it exists) is a term that has all the information in both the original terms, and an anti-unifier is a term that contains only all that is common between two terms. Also, notice that, depending on $\mathscr{L}$, anti-unifier and unifier might be unique or not.

1.4.1 Amalgams

The notion of *amalgam* can be conceived of as a generalisation of the notion of unification over terms. The unification of two terms (or descriptions) ψ_a and ψ_b is a new term $\phi \equiv \psi_a \sqcup \psi_b$, called unifier. All that is true for ψ_a or ψ_b is also true for ϕ; e.g., if ψ_a describes 'a red vehicle' and ψ_b describes 'a German minivan' then their unification yields the description 'a red German minivan.' Two terms are not unifiable when they represent incompatible or contradictory information; for instance 'a red French vehicle' is not unifiable with 'a blue German minivan'. The strict definition of unification means that any two descriptions with only one item with contradictory information cannot be unified.

An *amalgam* of two terms (or descriptions) is a new term that contains *parts from these two terms*. For instance, an amalgam of 'a red French vehicle' and 'a blue German minivan' would be 'a red German minivan'; clearly there are always multiple possibilities for amalgams, since 'a blue French minivan' is another possible amalgam. The notion of amalgam, as a form of 'partial unification', was formally introduced by Ontañón and Plaza (2010).

Definition 1.12 (Amalgam). The set of *amalgams* of two terms ψ_a and ψ_b is the set of terms such that:

$$\psi_a \curlyvee \psi_b = \{\phi \in \mathscr{L} \setminus \{\top\} \mid \exists \alpha_a, \alpha_b \in \mathscr{L} : \alpha_a \sqsubseteq \psi_a \wedge \alpha_b \sqsubseteq \psi_b \wedge \phi \equiv \alpha_a \sqcup \alpha_b\}$$

Thus, an amalgam of two terms ψ_a and ψ_b is a term that has been formed by unifying two generalisations α_a and α_b, whenever this unification is not inconsistent, i.e.,

[14] In Machine Learning, $A \sqsubseteq B$ usually means that A is more general than B, unlike in description logics, for instance, where it has the opposite meaning, since it is seen as 'set inclusion' of their interpretations.

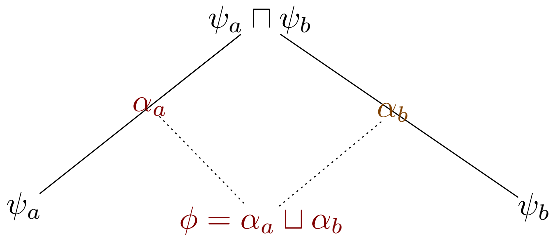


Fig. 1.2: A diagram of an amalgam ϕ from inputs ψ_a and ψ_b where $\chi = \alpha_a \sqcap \alpha_b$

$\alpha_a \sqcup \alpha_b \not\equiv \top$. Thus, an amalgam is a term resulting from combining some of the information in ψ_a with some of the information from ψ_b. Formally, $\psi_a \curlyvee \psi_b$ denotes the set of all possible amalgams; however, whenever it does not lead to confusion, we will use $\psi_a \curlyvee \psi_b$ to denote one specific amalgam of ψ_a and ψ_b.

Ontañón and Plaza (2010) give a slightly different definition of amalgam, for which not all generalisations are taken into account, only those that are less general than $\psi_a \sqcap \psi_b$ (the anti-unification of the inputs). We rephrase this definition here introducing the notion of *bounded amalgam*:

Definition 1.13 (Bounded amalgam). Let $\chi \in \mathscr{L}$. The set of χ-*bounded amalgams* of two terms ψ_a and ψ_b is the set of terms such that:

$$\psi_a \curlyvee_\chi \psi_b = \{\phi \in \mathscr{L} \setminus \{\top\} \mid \exists \alpha_a, \alpha_b \in \mathscr{L} : \chi \sqsubseteq \alpha_a \sqsubseteq \psi_a \wedge \chi \sqsubseteq \alpha_b \sqsubseteq \psi_b \wedge \phi \equiv \alpha_a \sqcup \alpha_b\}$$

A particularly interesting case (the one studied by Ontañón and Plaza (2010)) is when $\chi \equiv \psi_a \sqcap \psi_b$, the anti-unification of the inputs, as illustrated in Figure 1.2. The intuitive reason is that the anti-unification represents what is common or shared between the two inputs and, thus, generalising beyond $\psi_a \sqcap \psi_b$ would eliminate compatible information that is already present in both inputs.

The terms α_a and α_b are called the *transfers* or *constituents* of an amalgam $\psi_a \curlyvee \psi_b$. They represent all the information from ψ_a and ψ_b, respectively, which is *transferred* to the amalgam. As we will see later, this idea of transfer is akin to the idea of *transferring* knowledge from the source to target in CBR, and also in computational analogy (Falkenhainer et al., 1989).

Usually we are interested only in maximal amalgams of two input terms, i.e., those amalgams that contain maximal parts of their inputs that can be unified into a new coherent description. Formally, an amalgam $\phi \in \psi_a \curlyvee \psi_b$ is maximal if there is no $\phi' \in \psi_a \curlyvee \psi_b$ such that $\phi \sqsubset \phi'$. In other words, if more properties of an input were added, the combination would be no longer consistent. The reason why we might be interested in maximal amalgams is very simple: consider an amalgam ϕ' such that $\phi' \sqsubset \phi$; clearly ϕ', being more general than ϕ, has less information than ϕ and thus combines less information from the inputs ψ_a and ψ_b. Since ϕ has more information while being consistent, ϕ' or any amalgam that is a generalisation of ϕ, is trivially derived from ϕ by generalisation.

1.4.2 Asymmetric Amalgams and Analogy

There is a special case of amalgams of special interest: asymmetric amalgams, where the two input terms do not play a symmetrical role. The case of asymmetric amalgams, as we will show, is related to the notion of analogy and case-based inference, where one of the inputs (called the *source*) has much more information that the other input (called the *target* or *problem*). Asymmetric amalgams can be used to model the process by which knowledge from the source can be transfered to the target.

Definition 1.14 (Asymmetric amalgam). The χ-bounded *asymmetric amalgams* $\psi_s \overrightarrow{\curlyvee} \psi_t$ of two terms ψ_s (*source*) and ψ_t (*target*) is the set of terms such that:

$$\psi_s \overrightarrow{\curlyvee}_\chi \psi_t = \{\phi \in \mathscr{L} \setminus \{\top\} \mid \exists \alpha_s \in \mathscr{L} : \chi \sqsubseteq \alpha_s \sqsubseteq \psi_s \ \wedge\ \phi \equiv \alpha_s \sqcup \psi_t\}$$

In an asymmetric amalgam, the target term is transferred completely into the amalgam, while the source term is generalised. The result is a form of partial unification that retains all the information in ψ_t while relaxing ψ_s by generalisation and then unifying one of those more general terms with ψ_t itself. As before, we would be usually interested only in the asymmetric amalgams that are maximal.

This model of asymmetric amalgam can be used to model case-based inference in CBR, as explained in (Ontañón and Plaza, 2012), and analogical reasoning (Besold and Plaza, 2015; Besold et al., 2015). Essentially, this model clarifies what knowledge is transferred from source description to target, namely the transfer term α_s captures which case-based inference conjectures are applicable to (are consistent with) the target. In the case of a maximal amalgam, α_s represents as much information as can be transferred from the source to the target ψ_t such that $\alpha_s \sqcup \psi_t$ is consistent.

1.5 Relating Colimits and Amalgams

In Section 1.1 we mentioned that it is very appealing to model blending as a colimit in some category **C** of conceptual spaces and their structure-preserving mappings. When blending two input spaces, however, not everything is included into the blend because there may be incompatibilities between the input spaces. In general, conceptual blending is based on selective projections from the input spaces into the blend (Fauconnier and Turner, 2002).

Consequently, the classical colimit construct in **C** is inadequate for modelling blending. Goguen suggested $\frac{3}{2}$-colimits in ordered categories instead, where structure-preserving mappings between conceptual spaces are based on partial functions. We discussed this approach thoroughly in Section 1.3.

In Definition 1.6 we introduced an alternative way in which selective projection can be modelled categorically, without getting into the subtlety of dealing with ordered categories. In this section we shall focus on $\mathbf{Pfn}(\mathbf{C})$— the category of isomorphism classes of mono spans in $\mathbf{C}$—and show that the cocone constructs in $\mathbf{Pfn}(\mathbf{C})$ can be seen as an abstraction, into the category-theoretical setting, of amalgams as introduced in Section 1.4.1. Furthermore, this construct might be also suitable for modelling and computing conceptual blends, as we shall illustrate in Chapter 2. First, however, we recall some basic notions of category theory not introduced in Section 1.1 that we are going to need in this section, and we introduce also some additional notation.

1.5.1 Preliminaries

Let $\mathbf{C}$ be a category and $f\colon A \to C$ be a morphism in $\mathbf{C}$. We say that f *factors* through some morphism $g\colon B \to C$ if there exists $h\colon A \to B$ such that $f = h;g$. If g is a monomorphism, then h is the pullback of f along g.[15] Let $A \xrightarrow{f} C \xleftarrow{g} B$ be a diagram in $\mathbf{C}$. If there is a pullback over this diagram we shall write $\bar{f}$ for the pullback of morphism f along morphism g.

Remember from Definition 1.6 that a morphism $f : A \to B$ in $\mathbf{Pfn}(\mathbf{C})$ is, in particular, an isomorphism class of a span in $\mathbf{C}$. Without loss of generality, we will represent this class with a representative span $A \overset{f^-}{\longleftarrow\!\!\prec} A^0 \xrightarrow{f^+} B$. Recall that f^- is a monomorphism, i.e., the span is a mono span.

1.5.2 A Category-Theoretical Account of Amalgams

A poset $\langle \mathscr{L}, \sqsubseteq \rangle$ as the one considered in Section 1.4 can be seen as a category such that objects are the elements of $\mathscr{L}$, and there is a unique morphism from ϕ to ψ whenever $\phi \sqsubseteq \psi$. Consequently, we can propose a category-theoretical account of the notion of amalgam as given in Definitions 1.12 and 1.13.

Let $\mathbf{C}$ be a category and let C be an object in $\mathbf{C}$. We will say that the *generalisations* of C are all monomorphisms with target C. Let $f\colon A \to C$ be a morphism in $\mathbf{C}$. We will say that the f*-bounded generalisations* of C are all monomorphisms $g\colon B \rightarrowtail C$ such that f factors through g.

Now, let $\mathbf{C}$ be a category with pullbacks, and let $I_1 \xleftarrow{a_1} G \xrightarrow{a_2} I_2$ be a V-shaped diagram in the category $\mathbf{Pfn}(\mathbf{C})$ such that $a_1^- = a_2^- = id_G$. (Note that we can see it also as a V-shaped diagram $I_1 \xleftarrow{a_1^+} G \xrightarrow{a_2^+} I_2$ in $\mathbf{C}$.) Recall that for

[15] Following is a proof of this claim: Let $m\colon D \to A$ and $n\colon D \to B$ such that $m;f = n;g$. The morphism m is also the unique morphism from D to the apex A of the pullback such that $m;id_A = m$ and $m;h = n$. The first equality is trivial. For the second, we know that $m;f = n;g$ and $f = h;g$, consequently $m;h;g = n;g$. But g is a monomorphism, so $m;h = n$. And if k is any other morphism from D to the apex A satisfying these properties we would have that $k;id = m$, hence $k = m$.

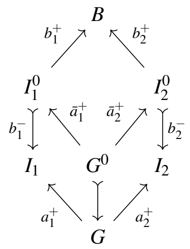


Fig. 1.3: Representation in $\mathbf{C}$ of a cocone in $\mathbf{MSpan}\,(\mathbf{C})$ over $I_1 \xleftarrow{a_1} G \xrightarrow{a_2} I_2$

$I_1 \xrightarrow{b_1} B \xleftarrow{b_2} I_2$ to be a cocone over this V-shaped diagram in $\mathbf{Pfn}(\mathbf{C})$ we need that $a_1;b_1 \simeq\!\!> a_2;b_2$. This amounts to saying that, in the $\mathbf{C}$-diagram of Figure 1.3, the pullbacks of $I_i^0 \xrightarrow{b_i^-} I_i \xleftarrow{a_i^+} G$ are isomorphic (G^0 denotes the apex of these isomorphic objects, without loss of generality), and $\bar{a}_1^+;b_1^+ = \bar{a}_2^+;b_2^+$. This brings us to the categorical notion of amalgam.

Definition 1.15 (Amalgam). Let $a_1^+ : G \to I_1$ and $a_2^+ : G \to I_2$ be two morphisms in a category $\mathbf{C}$ with pullbacks. An *amalgam* $\langle b_1^+, b_2^+ \rangle$ of a_1^+ and a_2^+ is a cocone with apex B over $I_1^0 \xleftarrow{\bar{a}_1^+} G^0 \xrightarrow{\bar{a}_2^+} I_2^0$, where $\bar{a}_i^+$ are the pullbacks of a_i^+ along generalisations $b_i^- : I_i^0 \to I_i$ of I_i (for $i \in \{1,2\}$), such that G^0 is the common (up to isomorphism) apex of these pullbacks (see Figure 1.3).

In the particular case when $\mathbf{C}$ is the poset $\langle \mathscr{L}, \sqsubseteq \rangle$ of Section 1.4 the definition above amounts to Definition 1.12 (taking as G the infimum element $\bot$). If we focus on a_i-bounded generalisations of I_i instead, we get Definition 1.13, where G plays the role of the element χ. This is so because in this case the apex G^0 of the pullback is isomorphic to G.

Definition 1.15 provides us a way to characterise conceptual blending in a manner that is faithful to the description given by Fauconnier and Turner (2002) and is independent of any particular choice of representation formalism for conceptual spaces and of any implementation thereof. Furthermore, the definition points to a possible way to compute blends via the classical colimit construct as implemented in HETS.

1.6 Conclusion

The theory of conceptual blending as put forward by Fauconnier and Turner in cognitive linguistics has been keenly adopted by researchers in the computing sciences for guiding the implementation of computational systems that aim at exhibiting creative capabilities, particularly when taking into consideration the invention of new concepts.

As is common with these early adoptions, each system has made its own choices of interpretation of the core elements that constitute Fauconnier and Turner's theory. They provide a formalisation of some fragment of theory that on the one hand attempts to be as faithful as possible to the intuitions stated by Fauconnier and Turner, and on the other hand would be feasible to implement in a computational system.

What has become evident from these early implementations of conceptual blending is that they have been designed in a very system-specific manner, without a clear separation of system-independent issues from those that are more system-specific. This makes it difficult to gain a deeper insight into the computational aspect of conceptual blending and hence to favour the reuse of blending technology to domains other than those envisioned by the system implementors.

In this chapter we have chosen to pursue a more domain- and system-independent approach to the development of a formal and computational theory of blending. In particular, we have taken the basic insight of Goguen that a blend might be adequately modelled as some kind of category-theoretical colimit, and we have expounded on the details of this insight in order to fully grasp its relationship with Fauconnier and Turner's theory.

Goguen himself proposed the framework of ordered categories to flesh out a mathematical account of conceptual blending, but he never fully worked out the implications of this proposal, nor did he show—other than with some small examples—how concrete acts of conceptual blending actually fit into his framework. The intuitions seemed convincing, but a thorough analysis was still missing. This is what we have started to do and what we have reported in this chapter.

What has become clear of our analysis is that dealing with Goguen's framework is much more subtle than originally expected. His notion of $\frac{3}{2}$-colimit as a way to model blending is quite complicated to grasp conceptually, in pasticular as a guide for the implementation of computational blending systems. Although the notion of colimit is, in our view, still a powerful notion to be exploited theoretically for the purpose of giving a precise characterisation of conceptual blending, we have considered alternative ways to do so, for instance, exploiting the notion of colimit in a category of spans. The advantage of such an approach is that it nicely covers also a generalisation of the notion of amalgam, originally proposed as a method for knowledge merging or integration in case-based reasoning. Indeed, the notion of amalgam is very reminiscent of that of blending, and by modelling blending as colimits in a category of spans we have become capable of bringing blending and amalgamation to the same theoretical footing.

The theoretical exploration carried out in this chapter will guide our subsequent work to carry out a computational realisation of blending that clearly distinguishes the domain-independent elements of blending such as amalgamation and colimit construction from the domain-specific realisations thereof. The uniformity provided by our model makes it possible to relate it with the mathematical model of the creative process proposed by Mazzola et al. (2011) and Andreatta et al. (2013). They propose to take the insights offered by the Yoneda lemma of category theory as a metaphor for the process by which an open question may be solved in a creative way. Schorlemmer et al. (2016) show by means of the Buddhist monk riddle (Koestler, 1964) that Mazzola et al.'s metaphor for the creative process can be useful to make explicit the external structure of the concept or idea we want to creatively explore. This metaphor likens the creative process to the task of finding a canonical diagram that externalises the structure of a categorical object. In particular we have focussed on the image-schematic structure in such a way that the solution to the riddle can be found by conceptual blending, using an amalgam-based process such as the one put forward in our model.

As future work, we intend to further explore our approach in other domains, validating the hypothesis that a relevant collection of image schemas should be sufficient to model diagrams that, via generalisation and colimit computation, yield novel and useful blends. Moreover, we surmise that for complex situations we will have not a blend but a web of blends, for example, situations where one or both input mental spaces are recursively blended. Such a web of blends is called Hyper-Blending Web (Turner, 2014). We intend to explore the span of the hypothesis that the input concepts in such a web of blends are image schemas and their specialisations, while the blend concepts are created by generalisation and colimit computation of image schemas and previous blends in the web.

References

M. Andreatta, A. Ehresmann, R. Guitart, and G. Mazzola. Towards a categorical theory of creativity for music, discourse, and cognition. In J. Yust, J. Wild, and J. A. Burgoyne, editors, *Mathematics and Computation in Music. 4th International Conference, MCM 2013. Montreal, QC, June 2013. Proceedings*, volume 7937 of *Lecture Notes in Artificial Intelligence*, 2013.

S. G. Baron and D. Osherson. Evidence for conceptual combination in the left anterior temporal lobe. *Neuroimage*, 55(4):1847–1852, 2011.

M. Barr and C. Wells. *Category Theory for Computing Science*. Prentice Hall International Series in Computer Science. Prentice Hall International, New York, 1990. ISBN 0-13-120486-6.

J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*, volume 44 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1997.

T. R. Besold and E. Plaza. Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, 2015.

T. R. Besold, K.-U. Kühnberger, and E. Plaza. Analogy, amalgams, and concept blending. In *Proceedings of the Third Annual Conference on Advances in Cognitive Systems*, 2015.

S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. The Millennium, 2012 update edition, 2012. URL http://www.math.uwaterloo.ca/~snburris/htdocs/ualg.html.

G. Calugareanu and I. Purdea. Examples in category theory, 2011. URL http://math.ubbcluj.ro/~calu/BOO-0-14.pdf. Unpublished book.

M. Codescu and T. Mossakowski. Heterogeneous colimits. In *Proceedings of the 2008 IEEE International Conference on Software Testing Verification and Validation Workshop*, ICSTW '08, pages 131–140, Washington, DC, USA, 2008. IEEE Computer Society.

M. Codescu, F. Horozal, M. Kohlhase, T. Mossakowski, F. Rabe, and K. Sojakova. Towards logical frameworks in the heterogeneous tool set HETS. In T. Mossakowski and H. Kreowski, editors, *Recent Trends in Algebraic Development Techniques — 20th International Workshop, WADT 2010, Etelsen, Germany, July 1–4, 2010, Revised Selected Papers*, volume 7137 of *Lecture Notes in Computer Science*, pages 139–159. Springer, 2010.

R. Diaconescu. *Institution-independent model theory*. Studies in Universal Logic. Birkhäuser Verlag, Basel, 2008. ISBN 978-3-7643-8707-5.

B. Falkenhainer, K. D. Forbus, and D. Gentner. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63, 1989.

G. Fauconnier. *Mental Spaces*. Cambridge University Press, 1994.

G. Fauconnier and M. Turner. Conceptual integration networks. *Cognitive Science*, 22(2):133–187, 1998.

G. Fauconnier and M. Turner. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books, 2002. ISBN 978-0-465-08785-3.

B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, 1999.

P. Gärdenfors. *Conceptual Spaces*. A Bradford Book, 2004.

R. W. Gibbs, Jr. Making good psychology out of blending theory. *Cognitive Linguistics*, 11(3–4):347–358, 2000.

J. Goguen. An introduction to algebraic semiotics, with applications to user interface design. In C. L. Nehaniv, editor, *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *Lecture Notes in Computer Science*, pages 242–291. Springer, 1999.

J. Goguen. Towards a design theory for virtual worlds: Algebraic semiotics and scientific visualization as a case study. In C. Landauer and K. Bellman, editors, *Proceedings Conference on Virtual Worlds and Simulation (Phoenix AZ, 7–11 January 2001)*, pages 298–303. Society for Modelling and Simulation, 2001.

J. Goguen. What is a concept? In F. Dau, M.-L. Mugnier, and G. Stumme, editors, *Conceptual Structures: Common Semantics for Sharing Knowledge. 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17–22, 2005. Proceedings*, volume 3596, pages 52–77. Springer, 2005a.

J. Goguen. Information integration in institutions. To appear in a memorial volume for Jon Barwise edited by L. Moss. Draft available at http://www.cs.ucsd.edu/users/goguen/pps/ifi04.pdf, 2005b.

J. Goguen. Mathematical models of cognitive space and time. In D. Andler, Y. Ogawa, M. Okada, and S. Watanabe, editors, *Reasoning and Cognition*, volume 2 of *Interdisciplinary Conference Series on Reasoning Studies*. Keio University Press, 2006.

J. A. Goguen. A categorical manifesto. *Mathematical Structures in Computer Science*, 1:49–68, 1991.

J. A. Goguen and D. F. Harrell. Style: A computational and conceptual blending-based approach. In S. Argamon, K. Burns, and S. Dubnov, editors, *The Structure of Style. Algorithmic Approaches to Understanding Manner and Meaning*. Springer, 2010.

J. Hayman and T. Heindel. On pushouts of partial maps. In H. Giese and B. König, editors, *Graph Transformation — 7th International Conference, ICGT 2014, Held as Part of STAF 2014, York, UK, July 22–24, 2014. Proceedings*, volume 8571 of *Lecture Notes in Computer Science*, pages 177–191. Springer, 2014.

P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium. 2 Volumes*, volume 44 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, Oxford, 2002.

W. Kahl. Collagory notes, version 1. SQRL Report 57, Software Quality Research Laboratory, McMaster University, 2010. URL `http://www.cas.mcmaster.ca/sqrl/papers/SQRLreport57.pdf`.

A. Koestler. *The Act of Creation*. Hutchinson & Co., 1964.

O. Kutz, T. Mossakowski, J. Hois, M. Bhatt, and J. Bateman. Ontological blending in DOL. In *Computational Creativity, Concept Invention, and General Intelligence — 1st International Workshop. International Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI-12), First, located at ECAI 2012, August 27, Montpellier, France*. Publication Series of the Institute of Cognitive Science, 2012.

O. Kutz, F. Neuhaus, T. Mossakowski, and M. Codescu. Blending in the Hub — towards a collaborative concept invention platform. In *Proceedings of the Fifth International Conference on Computational Creativity ICCC 2014*, 2014.

S. Lack. A 2-categories companion. In *Towards higher categories*, volume 152 of *IMA Vol. Math. Appl.*, pages 105–191. Springer, New York, 2010. DOI: 10.1007/978-1-4419-1524-5_4.

T. Leinster. A survey of definitions of *n*-category. *Theory Appl. Categ.*, 10:1–70 (electronic), 2002.

S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1998. ISBN 0-387-98403-8.

G. Mazzola, J. Park, and F. Thalmann. *Musical Creativity*. Computational Music Science. Springer, 2011.

C. McLarty. *Elementary Categories, Elementary Toposes*, volume 21 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, New York, 1992. ISBN 0-19-853392-6; 0-19-851473-5. Oxford Science Publications.

T. Mossakowski. Colimits of order-sorted specifications. In *Recent Trends in Algebraic Development Techniques (Tarquinia, 1997)*, volume 1376 of *Lecture Notes in Computer Science*, pages 316–332. Springer, Berlin, 1998.

T. Mossakowski, C. Maeder, and K. Lüttich. The Heterogeneous Tool Set. In O. Grumberg and M. Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems. 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 – April 1, 2007. Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, pages 519–522. Springer, 2007.

S. Ontañón and E. Plaza. Amalgams: A formal approach for combining multiple case solutions. In I. Bichindaritz and S. Montani, editors, *Case-Based Reasoning. Research and Development, 18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, July 19–22, 2010. Proceedings*, volume 6176 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2010.

S. Ontañón and E. Plaza. Toward a knowledge transfer model of case-based inference. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society (FLAIRS)*. AAAI Press, 2012.

F. C. Pereira. *Creativity and Artificial Intelligence: A Conceptual Blending Approach*, volume 4 of *Applications of Cognitive Linguistics*. Mouton de Bruyter, 2007.

B. C. Pierce. *Basic Category Theory For Computer Scientists*. Foundations of Computing Series. MIT Press, Cambridge, MA, 1991. ISBN 0-262-66071-7.

A. Poigné. Elements of categorical reasoning: Products and coproducts and some other (co-)limits. In D. Pitt, S. Abramsky, A. Poigné, and D. Rydeheard, editors, *Category Theory and Computer Programming (Guildford, 1985)*, volume 240 of *Lecture Notes in Computer Science*, pages 16–42. Springer, Berlin, 1986. DOI: 10.1007/3-540-17162-2_114.

E. Robinson and G. Rosolini. Categories of partial maps. *Information and Computation*, 79(2):95–130, 1988.

D. Sannella and A. Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012.

M. Schorlemmer and Y. Kalfoglou. Institutionalising ontology-based semantic integration. *Applied Ontology*, 3(3):131–150, 2008.

M. Schorlemmer, R. Confalonieri, and E. Plaza. The Yoneda path to the Buddhist monk blend. In *Proceedings of the Joint Ontology Workshops 2016 Episode 2: The French Summer of Ontology* co-located with the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016), Annecy, France, July 6–9, 2016.

J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.

P. Thagard. *Coherence in Thought and Action*. Life and Mind: Philosophical Issues in Biology and Psychology. MIT Press, 2000.

P. Thagard and T. C. Stewart. The AHA! experience: Creativity through emergent binding in neural networks. *Cognitive Science*, 35:1–33, 2011.

M. Turner. *The Origin of Ideas*. Oxford University Press, 2014.

T. Veale and D. O'Donoghue. Computation and blending. *Cognitive Linguistics*, 11 (3/4):253–281, 2000.