

A Uniform Model of Computational Conceptual Blending

Marco Schorlemmer and Enric Plaza

*Artificial Intelligence Research Institute, IIIA-CSIC
Bellaterra (Barcelona), Catalonia, Spain*

Abstract

We present a mathematical model for the cognitive operation of *conceptual blending* that aims at being uniform across different representation formalisms, while capturing the relevant structure of this operation. The model takes its inspiration from amalgams as applied in case-based reasoning, but lifts them into the theory of categories so as to follow Joseph Goguen’s intuition for a mathematically precise characterisation of conceptual blending at a representation-independent level of abstraction. We prove that our amalgam-based category-theoretical model of conceptual blending is essentially equivalent to the pushout model in the ordered category of partial maps as put forward by Joseph Goguen. But unlike Goguen’s approach, our model is more suitable for a computational realisation of conceptual blending, and we exemplify this by concretising our model to computational conceptual blends for various representation formalisms and application domains.

Keywords: conceptual blending, computational creativity, amalgams, category theory, case-based reasoning

1. Introduction

Fauconnier and Turner proposed *conceptual blending* as a fundamental cognitive operation underlying much of everyday thought and language [1, 2]. They described it as the process by which human cognition combines particular elements and relations of originally separate input mental spaces—“small conceptual packets constructed as we think and talk”—that share some common structure, into a blended space, in which new elements and relations emerge, and novel inferences can be drawn. Turner even goes a bit further and claims that our capacity for creating new ideas at a pace that completely outstrips any process of evolutionary change, and which ultimately has led to human culture, arises from our advanced ability to blend ideas, to make new ideas out of old ones, to do conceptual blending [3].

The cognitive, psychological and neural basis of conceptual blending has been extensively studied [2, 4, 5], 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 [6]. Nevertheless, the theory of conceptual blending has only been used in a constrained way for implementing creative computational systems. The reason for this is that Fauconnier and Turner did not aim at a computational model of conceptual blending, and thus they did not develop the sufficient details for this cognitive operation to be captured algorithmically. Still, their theory offers a useful framework for studying creative thinking using a computational approach and that is the reason why a number of researchers in the field of Computational Creativity have recognised the potential value of conceptual blending for guiding the implementation of creative systems [7–12].

These existing computational realisations of blending 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 [13], narrative style [14], poetry generation [15], or visual patterns [10]. 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. A major obstacle for a general account of computational conceptual blending is the lack of a mathematically precise theory that is suitable for the rigorous development of creative systems based on conceptual blending.

The original attempt to provide both a general and mathematically precise account of conceptual blending has been put forward by Joseph Goguen, initially as part of algebraic semiotics [16], and later in the context of a wider theory of concepts [17]. Based on the notion of ‘pushout’ in category theory, Goguen provided the core definitions of his theory and illustrated them on several examples, showing the aptness of this approach for formalising information integration [18] and reasoning about cognitive space and time [19]. As it stands, though, Goguen’s account is still quite abstract and lacks concrete algorithmic descriptions. Still, there are two reasons that make it an appropriate candidate theory on which to ground the formal model we are aiming at: First, it is an important contribution towards the unification of several formal theories of concepts, including the geometrical conceptual spaces of Gärdenfors [20], the symbolic mental spaces of Fauconnier [21], the information flow of Barwise and Seligman [22], the formal concept analysis of Ganter and Wille [23], and the lattice of theories of Sowa [24]. This makes it possible to potentially draw from existing algorithms that have already been developed in the scope of each of these frameworks. Second, 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 the input spaces of the blending process to be represented in a homogeneous manner across diverse domains. Current tools for heterogeneous specifications such as HETS [25] allow parsing, static analysis and proof management incorporating various provers and different specification languages.

In this article we present a novel mathematical theory of conceptual blending that has its origin in the notion of ‘amalgam’ in a lattice of generalisations as proposed in case-based reasoning (CBR) [26]. Amalgams model the combination of two cases into new ‘blended’ cases to be used in the CBR problem-solving process. As such, the notion of ‘amalgam’ seems to be related with, but not identical to, conceptual blends. Both are based on the idea of combining or fusing two or more different sources into a new entity that encompasses selected parts of the original sources; but they differ in the assumptions on the entities upon which they applied: amalgams combine ‘cases’ (expressed as terms in some formal language), while conceptual blends combine ‘mental spaces.’

The theory we propose here is a generalisation of the original notion of ‘amalgam’ from CBR, so as to constitute a valid alternative to Goguen’s mathematical theory of conceptual blending. It turns out, though, that our proposal actually has a deep connection with Goguen’s one, but unlike his original attempt, our amalgam-based theory is more amenable to be implemented in computational systems. By developing a formal, amalgam-based model of conceptual blending, we ultimately aim at providing general principles that may guide the design of computational 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 [27], semantic integration [28], or cognitive coherence [29]. With such a formal, but computationally feasible, model we shall contribute to further bridge the existing gap between the cognitive and theoretical foundations of conceptual blending, and its computational realisation.

A preliminary version of our model was developed in the context of the EU project COINVENT [30] as described in [31]. The present article, however, extends significantly the explanation to be found in that book chapter by presenting a more mature and detailed description of our mathematical model. First, we situate our model in the wider context of the various category-theory approaches that have been proposed for the study of human cognition (Section 2). Next, we give the basic intuitions behind modelling conceptual blending in category theory (Section 3) and provide our core model of conceptual blends as amalgams, lifted into the appropriate category (Section 4). Then, we show particular exemplifications of our framework that illustrate how blends are computed by means of amalgams (Section 5). Finally, we close the article with a discussion on related work (Section 6) and with our conclusions and reflections about possible future work (Section 7).

2. Category-Theory Approaches to Human Cognition

Category theory was initially developed out of algebraic topology with the objective of describing certain relationships between different mathematical entities [32]. Soon, however, its focus on structure-preserving relationships rather than on the structure of the entities under study proved the theory to be very

valuable as conceptual tool for modelling and relating many other fields of mathematics [33], and to further elucidate fundamental relationships of concepts beyond mathematics, in other scientific disciplines such as computer science [34] and physics [35].

The application of category theory to the life and cognitive sciences goes as far back as to Rosen’s proposal to model biological systems using the theory of categories and functors [36]. Halford and Wilson used the notion of ‘commutative diagram’ from category theory to define the appropriate application of symbolic processes to a particular segment of the environment at different stages of cognitive development [37]. Magnan and Reyes advocated for the use of category theory to adequately describe human cognitive processes, such as counting, as carried out by the human mind, in a way that avoids over-determination [38]. In particular they claimed that universals of the mind may be expressed by means of universal properties as postulated by the theory of categories.

The last decade has seen a significant growth in the number of scientific studies that have explored the suitability of category theory as conceptual tool for the cognitive sciences. In the spirit of Magnan and Reyes’s account, Phillips, Wilson and Halford have used categorical constructs with universal mapping properties to explain why certain cognitive abilities that seem different (such as transitive inference and class inclusion) turn out to be acquired at the same stage of cognitive development. They claim it is because they happen to be modelled with essentially the same kind of categorical construct—either a ‘product’ or a ‘coproduct’—and hence have the same structural complexity [39].

Furthermore, Phillips et al. highlight the value of the representation-independence that underlies this approach:

Category theory offers a potentially powerful approach to theorizing about cognition by not having to presuppose an, as yet, unknown internal structure for cognitive states representing task elements. [...] So, one is not required to make an a priori commitment to, say, symbolic or subsymbolic computational processes. In this sense, category theory complements more detailed (e.g., symbolic, or connectionist) approaches to cognitive modelling. [39]

Using a category-theoretic approach, Phillips and Wilson have also put forward a proposal to explain systematicity [40]—that cognitive capacities always exhibit certain symmetries—by means of the notion of ‘adjunction,’ so as to not require ad hoc assumptions as in classical or connectionist approaches [41–43]. An abstract category-theoretical viewpoint based on universal constructions also allowed them to establish a relationship with Gentner’s description of systematicity as used in computational models of analogy [44], and to present Fodor-Pylyshyn’s and Gentner’s accounts as “two sides of the same coin; two aspects of a common principle, universal construction,” thus claiming that such categorical construction is “a crucial component of cognitive architecture” [45]. As a consequence they also highlight the relationship with computation by way

of a categorical treatment of recursion to implement an associative learning method of systematicity [46].

Analogy has also been given a category-theoretic treatment by Arzi-Gonczarowski [47] in the particular context of artificial perception [48, 49]. It has recently been studied in more detail by Navarrete and Dartnell, who presented a category-theoretic model of analogy that uses the notion of ‘commutative diagram’ to study the effect of playing particular educational board games on the learning of numbers, that of ‘coequaliser’ to formally model the re-representation explaining ‘flexibility’ in analogy, and to further explain the acquisition of the concept of rational numbers in learning [50].

On the neuronal level of cognition, Healy, Caudell et al. have proposed mathematical models of cognitive neuroscience and neural networks based on category theory to address issues of human categorisation and similarity, and of episodic memory [51–53]; whereas Gómez-Ramírez and Sanz have used “neural categories” modelling the structural relationships and properties of place cells and grid cells in the brain’s hippocampus to study complex cognitive abilities such as memory, spatial perception and navigation by means of the notions of ‘coproduct’ and ‘colimit’ [54, 55]. Ellerman has also advocated for using a category-theoretic notion such as ‘adjunction’ to abstractly model the functions of perception and action of a brain [56–58].

Finally, Ehresmann and Vanbremeersch developed a category-theoretic model of Memory Evolutive Systems based on a hierarchy of complex components with multiple temporalities, such as biological, neuro-cognitive, or social systems [59]. The model is based on a theory of ‘dynamic’ categories, to capture the complexity, emergence and self-organisation of these systems. Ehresman and Gómez-Ramírez have further proposed this model to conciliate neuroscience and phenomenology [60].

The value of category theory in all these approaches to the modelling of human cognition stems from the fact that:

Category theory tries to uncover and classify the main operations of the “working mathematician;” [...] Mathematical activity, here, reflects some of the main operation that humans do for making sense of the world. [...] As all of these operations are at the root of our mental life, and also of science, it quite naturally follows that category theory can successfully applied to different scientific domains. [60]

3. Modelling Blends with Category Theory

3.1. Conceptual Blending

Fauconnier and Turner describe conceptual blending as the process by which two mental or conceptual spaces, whose structure can be put into correspondence via a cross-space relation—which in turn evokes a generic space of common structure—are combined to yield a new blended space, a composition done by

selectively projecting substructure of each of the input spaces into the blend and further adding emergent structure by a process of completion and elaboration.

The elements of blending are nicely exemplified by paying attention to the cognitive process described by Koestler [61] for solving a brain-teaser first put forward by Duncker [62], and later popularised by Gardner:

One morning, exactly at sunrise, a Buddhist monk began to climb a tall mountain. The narrow path, no more than a foot or two wide, spiralled around the mountain to a glittering temple at the summit. The monk ascended the path at varying rates of speed, stopping many times along the way to rest and to eat the dried fruit he carried with him. He reached the temple shortly before sunset. After several days of fasting and meditation he began his journey back along the same path, starting at sunrise and again walking at variable speeds with many pauses along the way. [...] Prove that there is a spot along the path that the monk will occupy on both trips at precisely the same time of day. [63]

Fauconnier and Turner [2] claim that, in the process recounted by Koestler of how a young woman managed to visualise the answer to the question posed by this brain-teaser, she took her mental space of the monk ascending the mountain on one day and the mental space of the same monk descending the mountain on a later day, and blended them into a new mental space in which the monk was at the same time ascending and descending the mountain:

I tried this and that, until I got fed up with the whole thing, but the image of that monk in his saffron robe walking up the hill kept persisting in my mind. Then a moment came when, super-imposed on this image, I saw another, more transparent one, of the monk walking down the hill, and I realized in a flash that the two figures must meet at some point some time — regardless at what speed they walk and how often each of them stops. Then I reasoned out what I already knew: whether the monk descends two days or three days later comes to the same; so I was quite justified in letting him descend on the same day, in duplicate so to speak. [61, p. 184]

Certain structure that the input spaces have in common is fused in the blend (the mountain, the path treaded by the monk, the start of the monk's journey at sunrise and its completion at sunset) while other elements are kept separate (the ascent from the mountain base to its summit and the descent from summit to base). Even the monk himself, which is actually the same entity in both input spaces, is *not* fused but kept separate, and two instances of the same monk occur in the blend, projected into it from each input space. Furthermore, not all structure present in the inputs is projected into the blended space (for instance, the specific days of the monk's ascent and descent or his speed of walking).

The solution to the problem arises in the blended space, where elements and structure of the input spaces have been put together. This composition

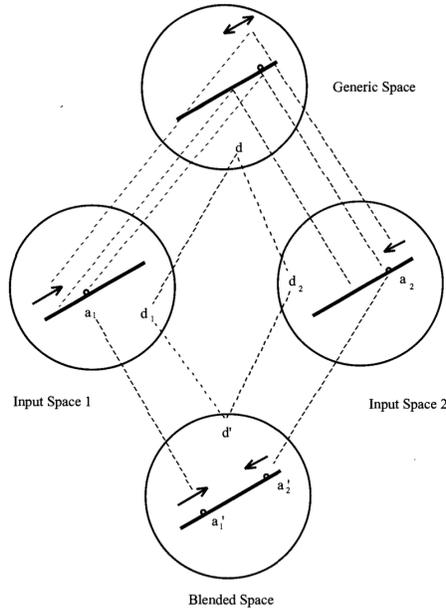


Figure 1: Schematic view of the conceptual blending that occurs when solving the problem of the Buddhist monk (taken from [2]). The generic space makes the structure shared by the inputs explicit (the path, the monk, the journey starting at sunrise and ending at sunset); the blended space has two instances of the same monk walking in opposite directions of the mountain path on the same day.

brings about new relationships—as the one between the two monks and their opposite directions of motion—and this further evokes our previous experiences of climbing up mountains and meeting fellow hikers on their way down. We thus complete the composition in the blend with the necessary structure of these experiences that allow us to infer that the ascending monk and the descending monk will meet at some time of the day at a particular spot on the path. This inference is effectuated by what Fauconnier and Turner call “running the blend”—the elaboration that eventually allows us to solve the problem (see Figure 1).

3.2. Category Theory

Following Goguen, we model conceptual blending using the theory of categories. In its application to computation, an important advantage of categorical approaches lies in them being independent of any particular implementation. For this reason, it is very appealing to search for a categorical framework in which to develop a computational theory of conceptual blending. Goguen initiated the development of such an approach to blending relying on the notion of ‘colimit’ by 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. [64, Section 6]

In Turner and Fauconnier’s account of conceptual blending the focus is on how little mental packages, mental frames, called ‘mental spaces’ form webs of conceptual connection between them. Little is specified about what sort of structure these mental spaces are. The whole emphasis in the description of conceptual blending is on the nature of the conceptual connections. Analogously, category theory puts the focus on the structure-preserving transformations (called morphisms, or arrows) between mathematical objects, not on the structure of the mathematical objects themselves.

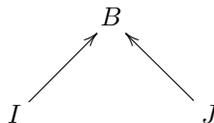
The basic intuition behind this modelling approach is to define conceptual blending as a particular category-theoretic construct that is as faithful as possible to the idea of blending. Mental spaces will correspond to objects in a category, and mappings and projections from one space to another will correspond to arrows. By framing blending in category theory, we will take the focus off the mental spaces and its structure and put it on the mappings and projections between spaces, and how these mappings relate to each other. Quoting Phillips and Wilson:

[Category theory] offers a re-conceptualization for cognitive science, analogous to the one that Copernicus provided for astronomy, where representational states are no longer the center of the cognitive universe—replaced by the relationships between the maps that transform them. [41]

In the following we will assume basic knowledge of category theory (see, for instance, [65]), although we will recall the core notions we use in this article.

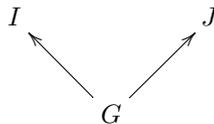
3.2.1. Diagrams, Cocones and Colimits

Let us focus on the basic blending scenario, that of blending two separate input spaces I and J . A very simple model of blending could be just two arrows, one from each of the input spaces I and J to a common blended space B that preserves the structure in I and J :

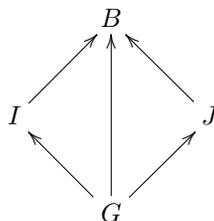


A simple pair of arrows, however, does not capture how the structure shared by input spaces is ultimately fused in the blended space, which is what makes blending such a powerful cognitive operation. Shared structure, as captured by a cross-space relation between input spaces, can be modelled categorically by

means of a *span*, a pair of arrows from a common generic space G —modelling the shared structure—to each of the input spaces:



Blends that fuse their common structure as captured by a span can then be modelled by *cocones*: we require that all arrows—obtained by composition—from the generic to the blended space are equal (we say the triangles in the diagram commute):

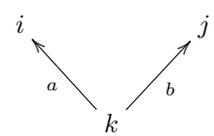


This idea can be generalised to any number of input spaces and arrows between them capturing their structural relationship.

Cocones are defined for particular diagrams. A *diagram* D in a category \mathcal{C} is a functor $D : \mathcal{I} \rightarrow \mathcal{C}$, where \mathcal{I} is a directed graph (where nodes are objects and edges are arrows in \mathcal{I}).¹ The graph \mathcal{I} is called the *shape* of D .

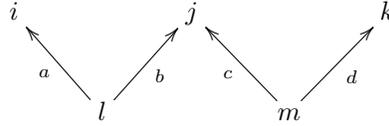
In other words, a diagram D consists of a family of \mathcal{C} -objects (indexed by the nodes of the graph \mathcal{I}) and a family of \mathcal{C} -arrows (indexed by the edges of \mathcal{I}) satisfying that for an edge a between nodes i and j , the associated \mathcal{C} -arrow Da has \mathcal{C} -objects Di and Dj as its source and target, respectively.

We are mostly interested in the case of finite diagrams, i.e., when the graph \mathcal{I} has a finite number of nodes and edges. In most such examples, instead of explicitly specifying the shape \mathcal{I} , we will simply draw a directed graph with the relevant \mathcal{C} -objects and \mathcal{C} -arrows. Two particular shapes of diagrams are going to be relevant in this article. When the shape is given by the graph



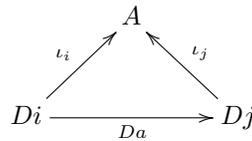
¹Strictly speaking \mathcal{I} is the free category generated over the directed graph, but for the purpose of this article it is not necessary to worry about this detail.

we called it a *v-diagram* (or *span*). If the shape is



we called it a *w-diagram*.

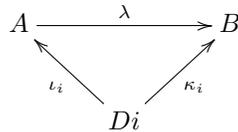
Definition 1 (Cocone) Let $D : \mathcal{I} \rightarrow \mathcal{C}$ be a diagram of shape \mathcal{I} in a category \mathcal{C} . A *cocone* for D is a \mathcal{C} -object A together with a family of \mathcal{C} -arrows $\{\iota_i : Di \rightarrow A\}$ (one for each \mathcal{I} -object i) such that, for each \mathcal{I} -arrow $a : i \rightarrow j$, the equality $Da; \iota_j = \iota_i$ holds —i.e., the following triangles commute:²



The \mathcal{C} -object A is called the *apex* of the cocone.

Of all these cocones, category theorists are interested in those ones that stand in relationship to all others, thus capturing the idea of a “minimal” cocone — one that, whilst capturing the structure of all objects of the diagram D and their relationships, does not add any new structure to it. Categorically this is expressed as follows: if there is another cocone (an alternative way of fusing common structure) then there is a unique arrow to this alternative cocone, making all triangles commute.

Definition 2 (Colimit) A cocone for D is said to be a *colimit* for D if, for each other cocone for D consisting of \mathcal{C} -object B together with the family of \mathcal{C} -arrows $\{\kappa_i : Di \rightarrow B\}$, there exists a unique \mathcal{C} -arrow $\lambda : A \rightarrow B$ such that, for each \mathcal{I} -object i , the equality $\iota_i; \lambda = \kappa_i$ holds —i.e., the following triangles commute:



When the diagram D is a span (a *v-diagram*), colimits are also called *pushouts*. When used to model blending, the apex of a pushout represents a space that includes the structure of the input spaces, and not more, fusing only the common structure specified in the generic space.

²We will use the diagrammatic notation ‘;’ for arrow composition more commonly used by computer scientists, instead of the functional notation ‘o’ more commonly used by mathematicians.

Cocones and pushouts seem to be the natural candidates to model blends; but they are not entirely adequate, because they do not capture selective projection (in a pushout *all* structure present in the input spaces is projected into the blend), emergent structure (in a pushout *no more* structure is added to the blend), nor incomparable alternative blends (a pushout is *unique* up to isomorphism).

3.2.2. Ordered Categories

Goguen suggested that ordered categories might be more adequate to approach the richness of conceptual blending. Ordered categories are categories whose sets of arrows from one object to another are partially ordered, providing a notion of ‘quality’ of arrows.

Definition 3 (Ordered Category) An *ordered category* is a category \mathcal{O} such that for every two objects A and B , there is a partial order \leq on $\text{Hom}(A, B)$ —the set of arrows from A to B — and composition is monotonic with respect to \leq in both arguments (i.e., if $f_1 \leq g_1$ and $f_2 \leq g_2$, then $f_1; f_2 \leq g_1; g_2$).

Unfortunately, in ordered categories the standard notions of limit and colimit appear to be inadequate because they do not preserve the order of arrows [66]. Consequently, a weaker notion has been proposed under the name of *near limit/colimit*, in which commutation (equality of arrows) is replaced by semi-commutation (inequality of arrows):

Definition 4 (Lax Cocone) Let $D : \mathcal{I} \rightarrow \mathcal{O}$ be a diagram of shape \mathcal{I} in an ordered category \mathcal{O} . A *lax cocone* for D is an \mathcal{O} -object A together with a family of \mathcal{O} -arrows $\{\iota_i : Di \rightarrow A\}$ (one for each \mathcal{I} -object i) such that, for each \mathcal{I} -arrow $a : i \rightarrow j$, the inequality $Da; \iota_j \leq \iota_i$ holds —i.e. the following triangles semi-commute:

$$\begin{array}{ccc}
 & A & \\
 \iota_i \nearrow & & \nwarrow \iota_j \\
 Di & \xrightarrow{Da} & Dj
 \end{array}$$

The universal property of a colimit needs to be replaced by asking for the arrow that is maximum among those satisfying the inequalities.

Definition 5 (Near Colimit) A lax cocone for D is said to be a *near colimit* for D if, for each other lax cocone for D consisting of \mathcal{O} -object B together with the family of \mathcal{O} -arrows $\{\kappa_i : Di \rightarrow B\}$, there exists an \mathcal{O} -arrow $\lambda : A \rightarrow B$ which is maximum for those satisfying that, for each \mathcal{I} -object i , $\iota_i; \lambda \leq \kappa_i$ —i.e.

satisfying that the following triangles semi-commute:

$$\begin{array}{ccc}
 A & \xrightarrow{\lambda} & B \\
 & \swarrow \iota_i & \searrow \kappa_i \\
 & D_i &
 \end{array}
 \leq$$

In particular, cocones and colimits in an ordered category are also lax cocones and near colimits. As before, when the diagram D is a span, near colimits are also called *near pushouts*.³ The following proposition lifting a well-known property from colimits to near colimits will be used later in Section 4.

Proposition 1 *The family of \mathcal{O} -arrows $\{\iota_i : Di \rightarrow A\}$ forming a near colimit for D is jointly epic.*

PROOF: Let $x, y : A \rightarrow B$ be a pair of \mathcal{O} -arrows. Suppose that $\iota_i; x = \iota_i; y$. This family of \mathcal{O} -arrows obviously forms a lax cocone for D . Consequently, there exists an \mathcal{O} -arrow $\lambda : A \rightarrow B$ which is maximum for those satisfying that $\iota_i; \lambda \leq \iota_i; x (= \iota_i; y)$. It follows that $x \leq \lambda$ and $y \leq \lambda$, and thus $\iota_i; x \leq \iota_i; \lambda$ and $\iota_i; y \leq \iota_i; \lambda$. Consequently, $\iota_i; \lambda = \iota_i; x = \iota_i; y$. Since λ is a maximum (and hence unique), it follows that $x = y$. \square

Goguen proposed near pushouts (which he calls $\frac{3}{2}$ -pushouts⁴) for modelling blends, in order to capture the additional posetal structure on arrows that ordered categories provide [16]. But modelling blends as near pushouts comes at a price: there can be several non-isomorphic near pushouts, i.e. several incomparable ideal blends; but Goguen perceived this as a strength rather than a weakness for modelling conceptual blending. From a computational perspective however, an important problem of this approach is that there is currently no computational support for ordered categories, while several computational tools for ‘plain’ categories have been developed that could potentially be used for computing blends, such as for instance the Heterogenous Tool Set [25].

4. Blends as Amalgams

The solution we propose is to revisit near pushouts in the context of categories of partial arrows—i.e., categories of spans consisting of a monic and a

³Jay in [67] initially also used the adjective ‘lax’ for the categorical constructs such as that of Definition 5, but it may be confused with more common uses of lax (co)limit as in [68]. In [66] he named them ‘local’, and in a subsequent extended unpublished manuscript he chose to use the adjective ‘near’ [69], instead. We settled in this article for the latter terminology, which is also the one adopted in [70].

⁴The reason for this name is because ordered categories are also called $\frac{3}{2}$ -categories. They are a special case of so-called 2-categories, where there is at most one 2-cell between two 1-cells (i.e., arrows); thus, ordered categories lie between plain 1-categories and 2-categories. Other names have also been used in the literature, such as locally partially ordered categories, locally posetal categories, Pos-enriched categories, and order-enriched categories.

total arrow, which is common practice in category theory. Partial arrows are the categorical abstraction of partial functions, which underly the selective projection occurring in conceptual blending. Furthermore, for modelling conceptual blending it seems that we do not need the general case of a near pushout, which would allow for partial arrows from the generic to the input spaces. First, let us introduce some additional category-theoretic notions.

4.1. Partial Arrows

A partial function can be characterised by a total function on the subset over which the partial function is defined. The categorical abstraction of this is to define a partial arrow $A \rightarrow B$ to be a span of two arrows $A \leftarrow A_0 \rightarrow B$ of which one is a mono (depicted with ‘ \rightarrow ’) representing the subobject A_0 . We call such a span a *monospan* and its subobject the *domain* of the partial arrow.

Strictly speaking, in a category \mathcal{C} , a subobject of an object A is an equivalence class of monos whose target is A and that factor through each other. It follows that the sources of the monos in such an equivalence class are isomorphic, and it is common to pick one of these isomorphic objects and to call it ‘the subobject’ of A . This applies also to partial arrows, which, strictly speaking, are to be understood as entire equivalence classes of monospans as determined by their respective subobjects.

For partial arrows to constitute a category we need to have inverse images (i.e., pullbacks of monos) to be able to define composition of partial arrows. Monos satisfying this property are called *stable* monos [66]. In the particular case of a pullback of two monos, the apex of the pullback is also referred to as the *intersection* of the subobjects represented by the monos, and given two subobjects A_1 and A_2 of an object A , we will write $A_1 \cap A_2$ for their intersection.

Following [71], we write $\text{Ptl}(\mathcal{C})$ for the category of partial arrows on \mathcal{C} .⁵ Often $\text{Ptl}(\mathcal{C})$ is too big because the class of monos of \mathcal{C} is too wide; thus, it is common to restrict the class of subobjects that are considered admissible as domains of partial arrows.⁶ We follow [66] and call such class a *realm* when its monos are stable and closed under composition and isomorphism; but we follow [71] and write $\mathcal{M}\text{-Ptl}(\mathcal{C})$ for the category of partial arrows, when the monos of the monospans representing its arrows are taken only from the realm \mathcal{M} .

When we move from total to partial functions, it is sensible to order functions by their extension. Hence homsets in the category of partial functions form a poset. In turn, homsets of partial arrows are also naturally ordered by the subobject ordering of their domains. This makes $\mathcal{M}\text{-Ptl}(\mathcal{C})$ an ordered category.

⁵There is a foundational issue to take into account here: homsets of $\text{Ptl}(\mathcal{C})$ are indeed sets iff the collections of subobjects for each object are sets.

⁶In categories such as **Set**, **Rng**, or **Grp** subobjects correspond to subsets, subrings, or subgroups; but they do not correspond to subspaces in **Top**, where one might consider as admissible domains only those monics that are continuous functions defined on an open subspace.

4.2. Amalgams

In spans modelling cross-space relations of conceptual blending (see Section 3.2.1), arrows are total in general, and they factor through the subobjects of the input spaces, which means that there exist also arrows from the generic space to the input subspaces (these subspaces lie in between the generic and the original input spaces, so to say) [31]. In this particular case, the near pushout diagram is significantly simplified, and it is essentially an abstraction of a what in [26] has been called an ‘amalgam’ in the context of case-based reasoning⁷ In the following we shall provide a category-theoretic abstraction of the original definition of ‘amalgam,’ keeping the totality condition on the arrows from the generic space, but not requiring them to factor through input subspaces.

Consequently, we need to define the generalisation of a v-diagram. For this we require that the subobjects of the realm \mathcal{M} have finite unions, and in the rest of this article we will assume that this additional condition holds for a class \mathcal{M} of monos to be considered a realm. The union of a family $\{A_i\}_{i \in I}$ of subobjects of A is defined as the subobject A' of A , denoted by $\bigcup_{i \in I} A_i$, which is preceded by each of the A_i (i.e., each A_i is also subobject of A'), and which has the following property: If, for an arrow $f : A \rightarrow B$, each A_i is carried into some subobject B' of B by f (i.e., there exists an arrow $f_i : A_i \rightarrow B'$ such that $m_i; f = f_i; n$, where m_i and n are the inclusion monos of A_i in A and B' in B , respectively), then A' is also carried into B' by f (i.e., there exists an arrow $f' : A' \rightarrow B'$ such that $m'; f = f'; n$, where m' is the inclusion mono of A' in A , denoted by $\bigcup_{i \in I} m_i$) [72].

Definition 6 (Generalisation of a v-diagram) Let \mathcal{C} be a category with realm \mathcal{M} . Let V be a v-diagram $I \xleftarrow{f} G \xrightarrow{g} J$ in \mathcal{C} . A *generalisation* of V is a w-diagram $I_0 \longleftarrow f^{-1}(I_0) \longmapsto f^{-1}(I_0) \cup g^{-1}(J_0) \longleftarrow g^{-1}(J_0) \longrightarrow J_0$ in \mathcal{C} , such that $m : I_0 \rightarrow I$ and $n : J_0 \rightarrow J$ are \mathcal{C} -monos in \mathcal{M} .

Notice that the objects of a generalisation of a v-diagram $I \longleftarrow G \longrightarrow J$ are all subobjects of either G , I or J , and that the diagram is determined by the pair of monic arrows $m : I_0 \rightarrow I$ and $n : J_0 \rightarrow J$. Consequently, we will also refer to a generalisation by means of this pair $\langle m, n \rangle$. The generalisations of a given v-diagram in a category \mathcal{C} with realm \mathcal{M} that has finite unions, together with the pairs $\langle v, w \rangle : \langle m, n \rangle \rightarrow \langle m', n' \rangle$ of monic arrows such that $m = v; m'$ and $n = w; n'$, form a category. It is straightforward to check that this category has finite intersections because the monos in \mathcal{M} are stable (see Section 4.1).

Definition 7 (Amalgam) Let \mathcal{C} be a category with realm \mathcal{M} . An *amalgam* for a v-diagram V in \mathcal{C} is the colimit for a generalisation W of V (see Figure 2).

When talking about a cocone for a w-diagram that is a generalisation of a v-diagram (see Definition 6), we will only explicitly talk of the injections

⁷This notion of ‘amalgam’ is not to be confused with the category-theoretic notion of ‘amalgamation’.

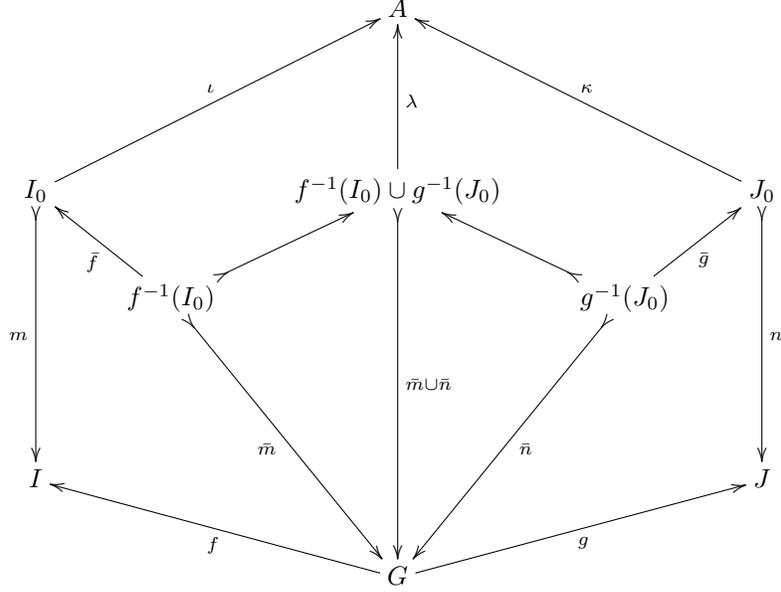


Figure 2: Amalgam with apex A . The \mathcal{C} -arrows ι , κ and λ are the injections of the colimit; the \mathcal{C} -monos m and n determine the generalisation of the ν -diagram; arrows \bar{m} and \bar{n} are the pullbacks of m and n along f and g , respectively; and $\bar{m} \cup \bar{n}$ is the inclusion mono of the union $f^{-1}(I_0) \cup g^{-1}(J_0)$ in G .

with sources I_0 , $f^{-1}(I_0) \cup g^{-1}(J_0)$ and J_0 . The other arrows of the cocone are determined by these and the arrows of the w -diagram by commutation (see Figure 2).

4.3. Amalgams vs. Near Pushouts

Kutz et al. formalised blending of ontologies following the same intuition of moving from arbitrary ordered categories to categories of partial arrows, and they suggested to model ontological blending by means of lax cocones of partial signature morphisms, which they called *blendoids* [73]. Amalgams are particular kinds of blendoids, satisfying a universal property, and in this subsection we show their relationship to near pushouts, Goguen's original mathematical model for blending as propounded in [16]. First, though, we need to state the following lemma:

Lemma 1 *Let \mathcal{C} be a category with realm \mathcal{M} . Let A together with \mathcal{M} -Pt(\mathcal{C})-arrows $(m, \iota) : I \rightarrow A$, $(n, \kappa) : J \rightarrow A$, and $(o, \lambda) : G \rightarrow A$ form a near pushout of the ν -shaped diagram $I \xleftarrow{(p,f)} G \xrightarrow{(q,g)} J$ in \mathcal{M} -Pt(\mathcal{C}); and let I_0 , J_0 and G_0 be the domains of partial arrows (m, ι) , (n, κ) and (o, λ) , respectively (i.e., they are the sources of \mathcal{C} -monos m , n and o). Necessarily the \mathcal{C} -object G_0 is isomorphic to the \mathcal{C} -object $f^{-1}(I_0) \cup g^{-1}(J_0)$.*

PROOF: Partial arrows (m, ι) , (n, κ) and (o, λ) form a near pushout. Consequently, $(p, f); (m, \iota) \leq (o, \lambda)$ and $(q, g); (n, \kappa) \leq (o, \lambda)$. Therefore, there are \mathcal{C} -monos from $f^{-1}(I_0)$ and from $g^{-1}(J_0)$ to G_0 , and hence there is also a \mathcal{C} -mono $\nu : f^{-1}(I_0) \cup g^{-1}(J_0) \rightarrow G_0$. Partial arrows (m, ι) , (n, κ) and $((\bar{m}; p) \cup (\bar{n}; q), \nu; \lambda)$ form a lax cocone for the same diagram, with \bar{m} and \bar{n} being the pullbacks along f and g , respectively, and $(\bar{m}; p) \cup (\bar{n}; q)$ being the inclusion mono of the union $f^{-1}(I_0) \cup g^{-1}(J_0)$. Consequently, there exists a maximum μ of those satisfying $(m, \iota); \mu \leq (m, \iota)$, $(n, \kappa); \mu \leq (n, \kappa)$ and $(o, \lambda); \mu \leq ((\bar{m}; p) \cup (\bar{n}; q), \nu; \lambda)$. It follows that there is a \mathcal{C} -mono from G_0 to $f^{-1}(I_0) \cup g^{-1}(J_0)$ capturing the partial order of the latter inequality, and hence $G_0 \cong f^{-1}(I_0) \cup g^{-1}(J_0)$. \square

The following theorem states that amalgams are “essentially” near pushouts for v-shaped diagrams of total arrows, provided we have images in the category we are working with. This holds for most categories in which we have computed conceptual blends (see Section 5). We say “essentially” because we relate these two constructs switching between a category \mathcal{C} and the ordered category $\mathcal{M}\text{-Ptl}(\mathcal{C})$ of its partial arrows (as equivalence classes of monospans).

Theorem 1 *Let \mathcal{C} be a category that has images, with realm \mathcal{M} . Let V be a v-diagram $I \xleftarrow{f} G \xrightarrow{g} J$ in the ordered category $\mathcal{M}\text{-Ptl}(\mathcal{C})$ of partial arrows of \mathcal{C} such that f and g are total. If $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -object A is the apex of an amalgam for V , then it is also the apex (taken as \mathcal{C} -object) of a near pushout for V .*

PROOF: Let W be the w-diagram that generalises V and for which A is the colimit. Take $\iota : I_0 \rightarrow A$, $\kappa : J_0 \rightarrow A$ and $\lambda : f^{-1}(I_0) \cup g^{-1}(J_0) \rightarrow A$ to denote the \mathcal{C} -arrows of this colimit, and $m : I_0 \rightarrow I$ and $n : J_0 \rightarrow J$ to denote the \mathcal{C} -monos that determine the generalisation W . It follows that $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrows (m, ι) , (n, κ) and $(\bar{m} \cup \bar{n}, \lambda)$ form a lax cocone for V , where \bar{m} and \bar{n} are the pullbacks of m and n along the \mathcal{C} -arrows f and g of V , respectively, and $\bar{m} \cup \bar{n}$ is the inclusion \mathcal{C} -mono of the union $f^{-1}(I_0) \cup g^{-1}(J_0)$.

Now, let $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrows (m', ι') , (n', κ') and (o', λ') be the injections of another lax cocone for V whose apex is B . Consequently, there is a w-diagram W' in \mathcal{C} that generalises V as determined by the \mathcal{C} -monos m' and n' and of which B is the apex of a cocone for W' . Take now the w-diagram $W \cap W'$ (see comment on page 14 following Definition 6) with its inclusion $\langle v, w \rangle : W \cap W' \rightarrow W$ and $\langle v', w' \rangle : W \cap W' \rightarrow W'$; and let us denote with $v \star w$ and $v' \star w'$ the monos between the union objects of the respective w-diagrams. Let A_0 denote the \mathcal{C} -object $Im(v; \iota) \cup Im(w; \kappa) \cup Im((v \star w); \lambda)$, and $u : A_0 \rightarrow A$ the inclusion \mathcal{C} -mono determined by this union of images.

The unique \mathcal{C} -arrows ι^* , κ^* and λ^* determined by the images $Im(v; \iota)$, $Im(w; \kappa)$ and $Im((v \star w); \lambda)$, respectively, form a cocone for $W \cap W'$ with apex A_0 . Furthermore, this cocone is a colimit for $W \cap W'$ because A with ι , κ and λ is a colimit for W , and A_0 is the smallest subobject of A for which $v; \iota$, $w; \kappa$ and $(v \star w); \lambda$ factor through. Since B is also an apex of the cocone for $W \cap W'$, with injections $v'; \iota'$, $w'; \kappa'$ and $(v' \star w'); \lambda'$, it follows that there is a unique arrow $\mu : A_0 \rightarrow B$ such that $\iota^*; \mu = v'; \iota'$, $\kappa^*; \mu = w'; \kappa'$ and $\lambda^*; \mu = (v' \star w'); \lambda'$.

The $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrow $(u, \mu) : A \rightarrow B$ is, thus, the maximal arrow satisfying $(m, \iota); (u, \mu) \leq (m', \iota')$, $(m, \kappa); (u, \mu) \leq (n', \kappa)$ and $(\bar{m} \cup \bar{n}, \lambda); (u, \mu) \leq (o', \lambda')$ which makes (m, ι) , (n, κ) and $(\bar{m} \cup \bar{n}, \lambda)$ a near pushout for V with apex A . \square

In general, not all near pushouts can be computed by means of amalgams (see Section 5). The following theorem, however, states that near pushouts are also amalgams when we are working in the context of a balanced category (i.e., arrows that are both monic and epic are also isomorphisms) that is also finitely cocomplete (i.e., finite diagrams have colimits):

Theorem 2 *Let \mathcal{C} be a finitely cocomplete and balanced category with realm \mathcal{M} . Let V be a v-diagram $I \xleftarrow{f} G \xrightarrow{g} J$ in the ordered category $\mathcal{M}\text{-Ptl}(\mathcal{C})$ of partial arrows of \mathcal{C} such that f and g are total. If \mathcal{C} -object A is the apex of a near pushout for V , then it is also the apex (taken as a $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -object) of an amalgam for V , provided that the generalisation of diagram V along $\langle m, n \rangle$ has a colimit.*

PROOF: Let $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrows $(m, \iota) : I \rightarrow A$, $(n, \kappa) : J \rightarrow A$ and $(o, \lambda) : G \rightarrow A$ be the partial arrows forming the near pushout for V . By the semi-commutativity of the lax cocone and by Lemma 1, \mathcal{C} -arrows ι , κ and λ form a cocone with apex A for the w-diagram W in \mathcal{C} that generalises V , as determined by the pair of monos $\langle m, n \rangle$. Let C be the apex of the colimit for W formed by \mathcal{C} -arrows ι' , κ' and λ' (which exists because of the finite cocompleteness of \mathcal{C}). Therefore there exists a unique arrow ξ such that $\iota'; \xi = \iota$, $\kappa'; \xi = \kappa$, and $\lambda'; \xi = \lambda$. In addition we have that C is also the apex of a lax cocone for V in $\mathcal{M}\text{-Ptl}(\mathcal{C})$ formed by $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrows (m, ι') , (n, κ') and $(\bar{m} \cup \bar{n}, \lambda')$. Since A is the apex of a near pushout, there exists a maximum $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrow $(u, \mu) : A \rightarrow C$ of those satisfying $(m, \iota); (u, \mu) \leq (m, \iota')$, $(n, \kappa); (u, \mu) \leq (n, \kappa')$ and $(\bar{m} \cup \bar{n}, \lambda); (u, \mu) \leq (\bar{m} \cup \bar{n}, \lambda')$.

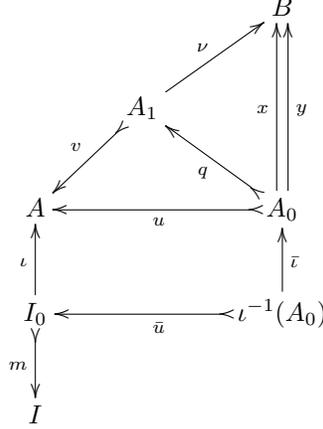
Let us focus on the first inequality (an analogous chain of reasoning can be followed with the other two): $(m, \iota); (u, \mu) \leq (m, \iota')$ implies that there is a monic arrow p such that the following diagram commutes:

$$\begin{array}{ccccc}
 & & A & \xleftarrow{u} & A_0 & \xrightarrow{\mu} & C \\
 & & \nearrow \iota & & \nearrow \bar{\iota} & & \nearrow \iota' \\
 I_0 & \xleftarrow{\bar{u}} & \iota^{-1}(A_0) & & & & \\
 \downarrow m & & \downarrow p & & & & \\
 I & \xleftarrow{m} & I_0 & & & &
 \end{array}$$

where \bar{u} and $\bar{\iota}$ are the pullback arrows for u and ι , respectively. Since m is monic, necessarily $p = \bar{u}$. Consequently, $\bar{u}; \iota' = \bar{\iota}; \mu$. Note that $\bar{\iota}$, $\bar{\kappa}$ and $\bar{\lambda}$ are jointly epic: let $x, y : A_0 \rightarrow B$ such that $\bar{\iota}; x = \bar{\iota}; y$, $\bar{\kappa}; x = \bar{\kappa}; y$, and $\bar{\lambda}; x = \bar{\lambda}; y$. The $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrows $(\bar{u}; m, \bar{\iota}; x)$, $(\bar{u}; n, \bar{\kappa}; x)$ and $(\bar{u}; \bar{m} \cup \bar{n}, \bar{\lambda}; x)$ form

a lax cocone with apex B for V in $\mathcal{M}\text{-Ptl}(\mathcal{C})$. Therefore, there exists a maximum $\mathcal{M}\text{-Ptl}(\mathcal{C})$ -arrow $(v, \nu) : A \rightarrow B$ of those satisfying $(m, \iota); (v, \nu) \leq (\bar{u}; m, \bar{t}; x)$, $(n, \kappa); (v, \nu) \leq (\bar{u}; n, \bar{\kappa}; x)$ and $(\bar{m} \cup \bar{n}\lambda); (v, \nu) \leq (\bar{u}; \bar{m} \cup \bar{n}, \bar{\lambda}; x)$.

Let us focus again on the first inequality. Since $(m, \iota); (u, x) = (m; \bar{u}, \bar{t}; x)$, we have that $(u, x) \leq (v, \nu)$. Necessarily there exists a monic arrow q such that $q; v = u$ and $q; \nu = x$:



An analysis similar to the one above can be done when composing with y instead of x . From all this and the fact that v is monic we can infer that $x = y$.

Recall that we concluded from $(m, \iota); (u, \mu) \leq (m, \iota')$ that $\bar{u}; \iota' = \bar{t}; \mu$. From this, and from $\iota'; \xi = \iota$ and the commutation of the pullback square, we deduce that $\bar{t}; \mu; \xi = \bar{t}; u$. Analogously, $\bar{\kappa}; \mu; \xi = \bar{\kappa}; u$ and $\bar{\lambda}; \mu; \xi = \bar{\lambda}; u$, and since \bar{t} , $\bar{\kappa}$ and $\bar{\lambda}$ are jointly epic, we deduce that $\mu; \xi = u$. We know that the arrow u is monic; but it is also epic, because by Proposition 1, we know that ι , κ and λ are jointly epic, and hence so is ξ . Since \mathcal{C} is balanced, u is an isomorphism, and we can conclude that ξ and $u^{-1}; \mu$ are inverse to each other, because C is the apex of the colimit for W . Therefore $A \cong C$. \square

5. Computing Blends With Amalgams

An advantage of using amalgams in place of near pushouts for modelling blends is that we do not need to work with an ordered category and can draw from the computational support available for computing colimits in plain, unordered categories. By lifting amalgams as originally proposed in [26] to category theory we get also a representation-independent framework.

To illustrate the constructs for modelling conceptual blending proposed in the previous section, we will show some examples of blends as modelled with amalgams in different categories: of typed feature structures [74], of \mathcal{EL}^{++} concept descriptions [75], and of CASL specifications [76]. The first of these examples, in Section 5.1, we describe in more detail than the remaining two,

because it exemplifies amalgams as put forward in this present article in its most general case (see Definition 7) and its relationship to near pushouts (see Theorem 1). The examples of Sections 5.2 and 5.3 illustrate a more constrained version of amalgam, bounded by the generic space of the v-diagram of the conceptual blend computed by the amalgam.

5.1. Blending Typed Feature Structures

As a first example we are going to illustrate the computation of blends of typed feature structures [74]. We briefly recall the basic definitions of this representation formalism.

Feature Structures

Feature structures are defined over a fixed finite set Feat of features and a fixed finite poset $\langle \text{Type}, \leq \rangle$ of types that is bounded-complete, i.e., all subsets of Type that have some upper bound also have a least upper bound, called *join*. In particular the join of the empty set is the *universal* type \perp , such that $\perp \leq \tau$, for all $\tau \in \text{Type}$.

Definition 8 (Feature Structure) A *feature structure* over Type and Feat is a tuple $F = \langle Q, \bar{q}, \theta, \delta \rangle$ where:

- Q is a finite set of nodes
- \bar{q} is a distinguished node in Q , called *root*
- $\theta : Q \rightarrow \text{Type}$ is a total *typing* function, assigning a type to each node in Q
- $\delta : \text{Feat} \times Q \rightarrow Q$ is a partial *feature value* function

satisfying that Q is *rooted* in \bar{q} : for all $q \in Q$, there exists a finite sequence of features f_1, f_2, \dots, f_n , with $n \geq 0$ (i.e., the sequence can be empty), such that $\delta(f_n, \dots \delta(f_2, \delta(f_1, \bar{q})) \dots) = q$.

We will use a graph notation to write feature structures as in [74], with roots identified by an unlabelled incoming arrow (see the icon blending example below).

Definition 9 (Feature-Structure Morphism) A *feature structure morphism* $h : F \rightarrow F'$ between feature structures $F = \langle Q, \bar{q}, \theta, \delta \rangle$ and $F' = \langle Q', \bar{q}', \theta', \delta' \rangle$ is a total function $h : Q \rightarrow Q'$ such that:

- $h(\bar{q}) = \bar{q}'$
- $\theta(q) \leq \theta'(h(q))$ for every $q \in Q$
- $h(\delta(f, q)) = \delta'(f, h(q))$ for every $q \in Q$ and every $f \in \text{Feat}$ for which $\delta(f, q)$ is defined

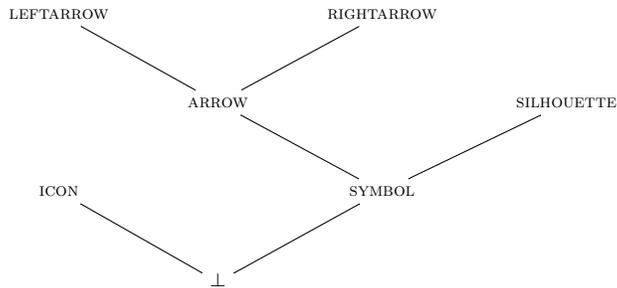
Feature structures over fixed **Type** and **Feat** together with feature structure morphisms form a category \mathcal{F} . Homsets in \mathcal{F} are either empty or singletons (because of the rootedness of feature structures), and when there is an \mathcal{F} -arrow from F to F' , it is said that F *subsumes* F' , and we write $F \sqsubseteq F'$. Consequently, \mathcal{F} is a preorder and all \mathcal{F} -arrows are both monic and epic. This implies that \mathcal{F} has images, because a mono is its own image; but, obviously, \mathcal{F} is not balanced because not all \mathcal{F} -arrows are isomorphisms.

Feature-structure unification is a pushout construction in \mathcal{F} , but since **Type** is only bounded-complete, \mathcal{F} is not cocomplete in general. Nevertheless, all non-empty subsets of **Type** have a greatest lower bound, called *meet*. Consequently, \mathcal{F} has pullbacks. Furthermore, **Type** is bounded-complete, and so is \mathcal{F} . Consequently, all subobjects of an object F do have an upper bound, namely F and, therefore, they have a least upper bound, which is their union. It follows that the class of \mathcal{F} -arrows has unions. All this makes the class of \mathcal{F} -arrows a *realm* and $\mathbf{Ptl}(\mathcal{F})$ and ordered category.

By Theorem 1, amalgams in \mathcal{F} are near pushouts in $\mathbf{Ptl}(\mathcal{F})$, and are thus suitable for modelling blends as represented with feature structures. But because \mathcal{F} is not a balanced category, there might be near pushouts in $\mathbf{Ptl}(\mathcal{F})$ that do not correspond to an amalgam in \mathcal{F} .

Icon Blending

As an example, consider feature structures describing icons composed by several symbols. Types of nodes are either **ICON** or represent symbols such as **SILHOUETTE** or **RIGHTARROW**, and features describe how symbols are positioned in the icon and with respect to other symbols in it. Let us assume the hierarchy **Type** is as follows (depicted as a Hasse diagram with the least informative universal type at a the bottom, and the most informative types in the leaves at the top):⁸



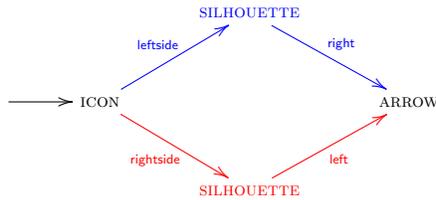
And let us assume that **Feat** includes the features *right*, *left*, *rightside* and *leftside*. In particular, let us consider the following feature structures representing

⁸This follows standard practice in domain theory, although it is contrary to how types and subtypes are depicted in knowledge representation. Consequently, the subsumption relation \sqsubseteq between feature structures is also written in the opposite direction to how it is usually done in description logics, for instance.

two icons that we would like to blend (colours identify which substructures of the feature structures correspond to the equally coloured fragments of the icons as depicted in Figure 3; they do not carry any meaning in feature structures):



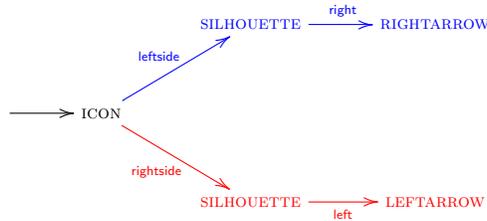
The structure shared by these feature structures is captured by the anti-unifier, which is going to play the role of the generic space of the blending process:



The colimit of the v-diagram formed by the two input feature structures and their anti-unifier, together with the subsumption morphisms from the anti-unifier to the inputs, if it existed, would correspond to the unifier of the input feature structures; but in this particular case there is no unifier because RIGHTARROW and LEFTARROW are not consistent: they do not have an upper bound in **Type**. However, there exist generalisations of the input feature structures—features structures subsuming the original inputs—that can be unified, for example:



The unifier of the generalised feature structures is:



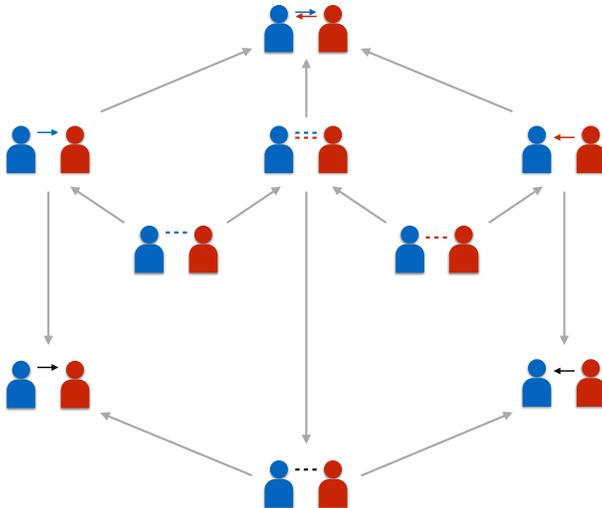


Figure 3: Amalgam of icons. Arrows are feature-structure morphisms (i.e., subsumptions) and point from subsumer to subsumee. A generic arrow (without direction) is drawn as a dashed line. Colours correspond to the substructures of the same colour in the feature structures as shown in the text.

Notice, though, that these generalised feature structures are *not* subsumed by the anti-unifier of the original input structures. Technically, this unifier is an amalgam as in Definition 7: it is the colimit of a w -diagram that generalises the v -diagram formed by the inputs and their anti-unifier (see Figure 3—where we have drawn the icons that might be represented by our feature structures—and compare it with Figure 2).

Amalgam Computation

There exists ample computational support for typed feature structures, particularly for computing feature-structure unification (i.e., pushout computation in \mathcal{F}) and feature-structure subsumption [77], since this representation formalism has been used for natural language processing [78], declarative programming [79, 80], conceptual modelling [81], case-based reasoning [82, 83], and music representation [84].

To generate blends of feature structures by way of computing amalgams, we need in addition to implement a generalisation refinement operator that allows us to explore the generalisation space of the original input spaces of the blend. By means of a simplistic Prolog implementation of the generalisation refinement operator described in [83], and encoding feature structures as open lists [85], we explored this generalisation space for our icon-blending example. Of all non-isomorphic amalgams generated by feature-structure unification on generalisations of the input feature structures, only four are maximal with re-

spect to \sqsubseteq and thus can be seen as the most informative blends we can obtain in this way; all others subsume either of these four.

5.2. Blending \mathcal{EL}^{++} Descriptions

Confalonieri et al. [86] have explored the invention of new concepts based on the principles of conceptual blending using description logics, the formalism that currently underlies most of the knowledge representation on the semantic web. In particular they have focused on the blending of concept descriptions in the lightweight ontology representation language \mathcal{EL}^{++} on which the OWL2 EL profile is based [86].

Unlike the example of Section 5.1, the notion of amalgam used in [86] is a constrained one, which is not equivalent to near pushouts. It is, however, more akin to the original definition of amalgam as provided by Ontañón and Plaza [26]. Let us introduce first this constrained understanding of amalgam. As in Definition 7, it is defined as the colimit of a generalisation of a given v-diagram $I \xleftarrow{f} G \xrightarrow{g} J$, only that G serves as an upper bound of the generalisations of inputs I and J . Furthermore, in [86] the definition is particular to \mathcal{EL}^{++} descriptions, and takes a common subsumer of I and J as the bound G . Following, we provide a definition for any category \mathcal{C} :

Definition 10 (Bounded Generalisation of a v-diagram) Let \mathcal{C} be a category with realm \mathcal{M} . Let V be a v-diagram $I \xleftarrow{f} G \xrightarrow{g} J$ in \mathcal{C} . A *bounded generalisation* of V is a v-diagram $I_0 \xleftarrow{f'} G \xrightarrow{g'} J_0$ in \mathcal{C} , such that $m : I_0 \rightarrow I$ and $n : J_0 \rightarrow J$ are \mathcal{C} -monos in \mathcal{M} that factor through f and g , respectively, i.e., $f'; m = f$ and $g'; n = g$.

Definition 11 (Bounded Amalgam) Let \mathcal{C} be a category with realm \mathcal{M} . A *bounded amalgam* for a v-diagram V in \mathcal{C} is the pushout for a bounded generalisation V' of V .

The monos m and n that characterise the generalisation V' of V factor through the respective arrows of V by definition, and consequently the objects $f^{-1}(I_0)$, $g^{-1}(J_0)$ and G happen to be isomorphic (and thus also $f^{-1}(I_0) \cup g^{-1}(J_0)$). Therefore, any bounded amalgam in \mathcal{C} is also an amalgam as in Definition 7, and thus also a near pushout in $\mathcal{M}\text{-PtI}(\mathcal{C})$; but, in general, the converse does not hold, unless *all* subobjects of I and J as characterised by monos in the realm \mathcal{M} factor through the arrows of v-diagram V . If this is the case, near pushouts are also bounded amalgams.

In [87], Confalonieri et al. provide an algorithm for computing conceptual blends out of input spaces specified in the description logic \mathcal{EL}^{++} . The algorithm computes bounded amalgams in the category \mathcal{E} of \mathcal{EL}^{++} descriptions over a fixed signature of concept, role and individual names, together with subsumption with respect to a fixed TBox \mathcal{T} (i.e., a finite set of axioms) as its arrows, going from

subsumer to subsumee; that is, there is an \mathcal{E} -arrow from C to D iff $C \sqsupseteq_{\mathcal{T}} D$.⁹ As with the category \mathcal{F} of typed feature structures given in Section 5.1, there is at most one arrow between any two objects, which is consequently both monic and epic.

Confalonieri et al. define a generalisation operator for \mathcal{EL}^{++} descriptions and implement the search process for bounded generalisations of input spaces by translating this operator into Answer-Set Programming (ASP). The operator is proper (it yields generalisations that are not equivalences) and locally finite (the number of generalisations generated for any given description is finite), but fails to be complete (there exists generalisations that are *not* generated by the operator). Actually, no locally finite, proper and complete generalisation operator exists for \mathcal{EL}^{++} descriptions. Furthermore, in order to avoid the generation of infinite chains of generalisations, the number of nested existential quantifiers is limited to a maximum $k \in \mathbb{N}$, the so called *role depth* of \mathcal{EL}^{++} descriptions.

Consequently, the focus is actually on a full subcategory \mathcal{E}_k of \mathcal{E} , whose objects are \mathcal{EL}^{++} descriptions of role depth $\leq k$. The realm \mathcal{M} underlying the amalgam computation described in [87] is, thus, the set of \mathcal{E}_k -arrows (i.e., of subsumptions $\sqsupseteq_{\mathcal{T}}$) that can be effectively generated by the generalisation operator. \mathcal{M} is indeed a realm because its monos are stable and closed under composition and isomorphism. In particular, stability follows from the fact that unique (up to equivalence) role-depth bounded least common subsumers exist for every pair of \mathcal{EL}^{++} descriptions; this has been shown for several variants of \mathcal{EL} -based languages [88–90], and can be easily extended to \mathcal{EL}^{++} .

The bounded amalgams are then computed by taking the greatest common subsumee of the generalised concept descriptions, which in \mathcal{EL}^{++} is just their intersection (\sqcap). The space of potential blends is further pruned by additional constraints, and partially ordered according to some preference criteria that attempt to capture the fact that conceptual blending is usually realised with some objective in mind, and driven by the values of particular audiences [91]. For further details about the computational realisation of conceptual blending of \mathcal{EL}^{++} descriptions we refer to [87].

5.3. Blending CASL Specifications

We conclude our examples of blends as modelled with amalgams in different categories by paying attention now to a more expressive representation formalism than those of the previous two subsections, and we illustrate our mathematical framework with spaces characterised as CASL specifications.

CASL is the Common Algebraic Specification Language developed by the Common Framework Initiative (CoFI). It is a general-purpose specification language based on first-order logic with induction. In [92], Eppe et al. have used CASL to specify chords with the aim of generating novel cadences and chord progressions by conceptual blending. (For example, conceptual blending between

⁹Recall that the subsumption relation between DL descriptions is written in the opposite direction to how it is done between feature structures.

the perfect and Phrygian cadence gives rise to the tritone substitution progression and the backdoor progression.) As with blends of \mathcal{EL}^{++} descriptions, only bounded amalgams are computed, and chord generalisation is implemented by means of an ASP program that codes a generalisation operator for CASL specifications. A very simple and straightforward generalisation operator is defined in [92], which just takes a proper subset of the specification by removing one axiom of a given specification.

The use of a general-purpose specification language such as CASL broadens the possible domains on which to apply amalgam-based conceptual blending computation. In [93], Eppe et al. report on a blending framework that illustrates the blending of the mathematical theories of natural numbers and of lists as specified in CASL with the objective to provide computational support in the discovery of so-called *Eureka lemmas*, i.e., intermediate lemmas discovered during the exploration of a theory that can be used to help with the proof obligations during this exploration.

In [94] amalgam-based blending of CASL theories is also applied to the computation of cross-domain blends, where input spaces are taken from two separate domains, such as for example chord progressions on the one hand and mathematical theories on the other hand, with the aim of generating very creative and surprising blends. Formalising blending in a representation-independent way allows Eppe et al. to computationally explore the invention of such multi-domain concepts. For example, by blending the specification of cyclic groups of elements with a specification for chords defined via intervals between their notes, Eppe et al. are capable of reproducing a very creative technique in modern jazz, namely *voicing*; and they are also capable of generating various kinds of chord progressions, including a progression known in jazz as *Coltrane Changes* [94].

6. Related Work

We mentioned in the introduction that several computer implementations of conceptual blending have been proposed, particularly in the field of Computational Creativity for guiding the development of creative systems. Although they did not aim at providing a representation-independent, uniform model of conceptual blending, each of them attempts to concretise Fauconnier and Turner’s basic insights in more or less degree, providing some sort of formalisation of the basic constitutive elements of conceptual blending and the optimality principles that govern good blends as proposed by Fauconnier and Turner.

Sapper [7]. Originally developed as a computational model of metaphor and analogy, in order to compute a mapping between two separate domains that respects the relational structure of concepts in each domain. Strictly speaking, Sapper does not generate blends, but structural mappings between input concepts. It can, however, be seen as a computational model for conceptual blending, because the mappings between the pairs of concepts that constitute its output can be manipulated as atomic units, as blended concepts. Furthermore, Sapper does not work with a priori given input spaces. It is the structure

mapping algorithm itself which, given two domains to be mapped, determines the set of concepts and relations between these concepts that constitute the spaces that are blended. It does so by searching within its semantic memory for the largest substructures (bounded by a previously fixed size) at the root concepts of these domains that are isomorphic as with respect to their relational structure, and whose concepts are metaphorically related. Semantic memory is represented as a semantic network, a graph whose nodes represent concepts and whose edges represent binary relations between concepts. The actual correspondence or blend between concepts is computed by applying a spreading activation algorithm according to which Sapper locates pairs of paths that are structurally isomorphic (of equal length and constituted by the same sequence of semantic relations) and that terminate at concepts that Sapper considers semantically bridgeable. Building upon this work, Metaphor-Eyes is a web-based approach to generate more loosely defined blends [95]. Inconsistencies and evaluation metrics are not addressed, and, instead, Metaphor-Eyes takes the Internet as a provider for the massive amount of background information that is required to generate meaningful blends. Sapper has subsequently been used in a more goal-driven selective projection process following Brandt and Brandt's theory of conceptual blending [96] for generating novel, standalone concepts such as fictional gadgets in computer-generated stories, or objects used in pretend play that combine features of a desired fantasy-world object with a real-world object at hand [11].

Alloy [97]. This algorithm for conceptual blending incorporates many ideas of the algebraic semiotics approach by Goguen [16] and of conceptual blending theory. Alloy has been integrated in the Griot system for automated narrative generation [15, 98]. Apart from the primary conceptual blending approach realised with Alloy, Griot also uses a secondary structural blending mechanism that blends the dynamic elements of natural language narratives to generate poetry. The input spaces of the Alloy algorithm are theories defined in the algebraic specification language BOBJ [99]. This allows one to represent sorts, operators, constants and axioms. The blending algorithm generates two binary trees that are based on the input graph, and which represent (i) the space of possible mappings of relations and (ii) the space of possible mappings of constants, respectively. These trees are then combined in the sense that the leaves of the tree of constants are applied to the trees of relations. The resulting combined tree has leaves that represent all possible sort-preserving mappings of relations and constants, i.e., all possible blends. During the tree generation, certain optimality principles are applied to prune the space.

Divago [8, 100, 101]. Probably the first complete implementation of conceptual blending. Its knowledge base contains different micro-theories and their instantiations. Of these, two are selected for blending, either by the user or randomly. A mapper then generates the generic space between the inputs, and passes it to a blender module which generates the blendoid, i.e., a projection that defines the space of possible blends. Blending in Divago is done on pairs of concepts,

taken from the multi-domain knowledge base of the system. A factory component is used to select the best blends among the blendoid by means of a genetic algorithm. A dedicated module implements the optimality principles. Given a blend, this module computes a measure for each principle. These measures yield a preference value of the blend that is taken as the fitness value of the genetic algorithm. Recent work [102] describe the use of Divago as a blending component within a computational creativity infrastructure, and discuss the relevance of the optimality principles in the Divago’s architecture.

HDTP [103]. Originally conceived as a framework for analogical reasoning [104], using a many-sorted first order language to represent conceptual spaces. In HDTP-based analogical reasoning, knowledge is mapped and transferred from a usually well-known source domain to a target domain. This happens in two phases. In the mapping phase, source and target are compared to find commonalities. In particular, HDTP uses second-order anti-unification, which is restricted in a way that renders the process decidable. In the transfer phase, unmatched knowledge in the source is mapped to the target to establish new hypotheses. The blending of two initial theories is done in three steps. First, core blend laws are applied, second, preferred conjectures are added, and third, extra conjectures are added.

In a subsequent development, an HDTP-based theory blending algorithm has been proposed for the generation of novel mathematical theories [105]. It computes maximally informative and maximally compressed theory blends by exploring the space of possible blends of two input theories relative to a common generalisation of these computed by HDTP. The algorithm is devised as an interactive process where the user evaluates the outputs and decides to trigger a relaxation of the common generalisation in order to determine a new space of possible blends to explore.

Cobble [106]. Developed in the context of the European COINVENT project [30], this prototype is based on some early ideas of theoretical and computational principles put forward in this article. It is a flexible and modular concept invention prototype implemented and encapsulated as RESTful services. Cobble instantiates the amalgam-based concept blending model and implements a workflow for concept invention. Input spaces are specified using DOL [107], an international ontology interoperability standard that provides a unified metalanguage for employing an open-ended number of formal logics. Generic space computation is achieved either using HDTP or Answer-Set Programming (ASP). ASP is also the technique used to search the generalisation spaces of inputs. Colimit computation is done using HETS [25], a parsing, static analysis and proof management tool incorporating various provers and different specification languages, thus providing a tool for heterogeneous specifications.

7. Conclusion

We have proposed to take amalgams as originally used for case-based reasoning to act as a model for conceptual blending. To obtain a mathematically

precise, uniform model that is applicable to a wide range of different representation formalisms, we have opted for a category-theoretic definition of amalgam, drawing from the theory of ordered categories and partial arrows. This abstract approach situates our work within recent efforts to use category theory for modelling and reasoning about the structure and operations of human cognition.

Amalgams, as we define them category-theoretically, lie between pushouts (which are too restrictive for modelling blends) and cocones (which cover too many irrelevant cases). They are “essentially equivalent” to near pushouts (Goguen’s $\frac{3}{2}$ -pushouts) of cross-space relations in the corresponding ordered category of partial arrows, but unlike near pushouts, they are more suitable for a computational realisation of conceptual blending. To illustrate this, we have discussed three different representations of computational conceptual blending as uniformly implemented with amalgams.

Also based on Goguen’s work on $\frac{3}{2}$ -pushouts in ordered categories, there is a recent effort by Diaconescu to extend the category-theoretic notion of ‘institution’—widely used in formal specification—to encompass ordered categories, with the explicit aim of providing adequate mathematical foundations to conceptual blending [70]. He argues that such foundations necessarily need to include a proper treatment of the semantic dimension by means model amalgamation in lax cocones. However, by sticking, as Goguen did, to the ordered categorical framework, his proposal is still computationally difficult to realise. In our future work we want to explore how our amalgam-based approach for computational conceptual blending relates to $\frac{3}{2}$ -institutions and model amalgamation.

The representation-independent character of our category-theoretic model will allow us to establish interesting connections with other related frameworks that we would like to explore. In [108], for instance, we link core aspects of certain creative problem-solving processes based on conceptual blending to amalgams for diagrams of *image schemas*—basic skeletal patterns that recur in our sensory and motor experience [109, 110]—by framing them in terms of the category-theoretic model of the creative process put forward by Mazzola et al., which is based on the Yoneda Lemma [111, 112]. We think that by means of our model we could further provide mathematically rigorous definitions of ‘conceptual metaphor’, ‘metaphorical entailment’ and ‘metaphor aptness’ [113] by drawing from Barwise and Seligman’s work on ‘distributed logics’ and ‘reasoning at a distance’ using channel theory [22].

Finally, we think that the uniform model presented in this article can also be very relevant for further understanding the relationship of embodied cognition with conceptualisation and creativity in computational environments. While Hedblom et al. have done some initial work in this direction by developing an *image schema logic* [114], we think that by building upon a category-theoretic approach we can reach a uniform but computationally feasible model of the image-schematic structure underlying human conceptualisation and creative-problem solving that is applicable to a variety of different domains.

Acknowledgments

This research was supported by the COINVENT project funded by the Seventh Framework Programme for Research of the European Commission, under FET-Open grant no. 611553; by the CSIC *Proyecto Intramural Especial DIVERSIS*, under grant no. 201750E064; and by the *Ajuts a grups de recerca consolidats* from the *Generalitat de Catalunya*, under grant no. 2017 SGR 172.

References

- [1] G. Fauconnier, M. Turner, Conceptual integration networks, *Cognitive Science* 22 (1998) 133–187.
- [2] G. Fauconnier, M. Turner, *The Way We Think*, Basic Books, 2002.
- [3] M. Turner, *The Origin of Ideas*, Oxford University Press, 2014.
- [4] R. W. Gibbs, Jr., Making good psychology out of blending theory, *Cognitive Linguistics* 11 (2000) 347–358.
- [5] S. G. Baron, D. Osherson, Evidence for conceptual combination in the left anterior temporal lobe, *Neuroimage* 55 (2011) 1847–1852.
- [6] M. Turner, Blending and conceptual integration, <http://markturner.org/blending.html>, n.d. Last accessed on Dec 12, 2019.
- [7] T. Veale, D. O’Donoghue, Computation and blending, *Cognitive Linguistics* 11 (2000) 253–281.
- [8] F. C. Pereira, Creativity and Artificial Intelligence: A Conceptual Blending Approach, volume 4 of *Applications of Cognitive Linguistics*, Mouton de Bruyter, 2007.
- [9] J. A. Goguen, D. F. Harrell, Style: A computational and conceptual blending-based approach, in: S. Argamon, K. Burns, S. Dubnov (Eds.), *The Structure of Style. Algorithmic Approaches to Understanding Manner and Meaning*, Springer, 2010, pp. 291–316.
- [10] P. Thagard, T. C. Stewart, The AHA! experience: Creativity through emergent binding in neural networks, *Cognitive Science* 35 (2011) 1–33.
- [11] B. Li, A. Zook, N. Davis, M. O. Riedl, Goal-driven conceptual blending: A computational approach for creativity, in: M. L. Maher, K. J. Hammond, A. Pease, R. Pérez y Pérez, D. Ventura, G. A. Wiggins (Eds.), *Proceedings of the Third International Conference on Computational Creativity (ICCC 2012)*, 2012, pp. 9–16.

- [12] M. Žnidaršič, A. Cardoso, P. Gervás, P. Martins, R. Hervás, A. O. Alves, H. G. Oliveira, P. Xiao, S. Linkola, H. Toivonen, J. Kranjc, N. Lavrač, Computational creativity infrastructure for online software composition: A conceptual blending use case, in: F. Pachet, A. Cardoso, V. Corruble, F. Ghedini (Eds.), Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016), Sony CSL, Paris, France, 2016, pp. 371–378.
- [13] J. Goguen, Semiotic morphisms, representations and blending for interface design, in: F. Spoto, G. Scollo, A. Nijhost (Eds.), Algebraic Methods in Language Processing. Proceedings Twente Workshop on Language Technology 21, Universiteit Twente, Enschede, Netherlands, 2003, pp. 1–15.
- [14] B. Dancygier, Blending and narrative viewpoint: Jonathan raban’s travels through mental spaces, *Language and Literature* 14 (2005) 99–127.
- [15] D. F. Harrell, Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation, in: Proceedings of the 6th Digital Arts and Culture Conference (DAC 2005), 2005, pp. 133–143.
- [16] J. Goguen, An introduction to algebraic semiotics, with applications to user interface design, in: C. L. Nehaniv (Ed.), Computation for Metaphors, Analogy, and Agents, volume 1562 of *Lecture Notes in Computer Science*, Springer, 1999, pp. 242–291.
- [17] J. Goguen, What is a concept?, in: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17-22, 2005. Proceedings, volume 3596 of *Lecture Notes in Artificial Intelligence*, Springer, 2005, pp. 52–77.
- [18] J. Goguen, Information integration in institutions, Unpublished manuscript available at <http://cseweb.ucsd.edu/~goguen/pps/ifi04.pdf>, n.d. Last accessed on Dec 12, 2019.
- [19] J. Goguen, Mathematical models of cognitive space and time, in: D. Aandler, Y. Ogawa, M. Okada, S. Watanabe (Eds.), Reasoning and Cognition, volume 2 of *Interdisciplinary Conference Series on Reasoning Studies*, Keio University Press, 2006, pp. 125–128.
- [20] P. Gärdenfors, *Conceptual Spaces*, A Bradford Book, 2004.
- [21] G. Fauconnier, *Mental Spaces: Aspects of Meaning Construction in Language*, Cambridge University Press, 1994.
- [22] J. Barwise, J. Seligman, Information Flow: The Logic of Distributed Systems, volume 44 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 1997.

- [23] B. Ganter, R. Wille, *Formal Concept Analysis*, Springer, 1999.
- [24] J. F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, 2000.
- [25] T. Mossakowski, C. Maeder, K. Lüttich, The Heterogeneous Tool Set, in: O. Grumberg, M. Huth (Eds.), *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*, Springer, 2007, pp. 519–522.
- [26] S. Ontañón, E. Plaza, Amalgams: A formal approach for combining multiple case solutions, in: *CCBR'10: 18th International Conference on Case-Based Reasoning*, volume 6176 of *Lecture Notes in Artificial Intelligence*, Springer, 2010, pp. 257–271.
- [27] B. Falkenhainer, K. D. Forbus, D. Gentner, The structure-mapping engine: Algorithm and examples, *Artificial Intelligence* 41 (1989) 1–63.
- [28] M. Schorlemmer, Y. Kalfoglou, Institutionalising ontology-based semantic integration, *Applied Ontology* 3 (2008) 131–150.
- [29] P. Thagard, *Coherence in Thought and Action, Life and Mind: Philosophical Issues in Biology and Psychology*, MIT Press, 2000.
- [30] M. Schorlemmer, A. Smaill, K.-U. Kühnberger, O. Kutz, S. Colton, E. Cambouropoulos, A. Pease, COINVENT: Towards a computational concept invention theory, in: S. Colton, D. Ventura, N. Lavrač, M. Cook (Eds.), *Proceedings of the Fifth International Conference on Computational Creativity (ICCC 2014)*, Ljubljana, Slovenia, 9th – 13th June 2014, Jožef Stefan Institute, Ljubljana, Slovenia, 2014, pp. 288–296.
- [31] F. Bou, E. Plaza, M. Schorlemmer, Amalgams, colimits, and conceptual blending, in: [106], 2018, pp. 3–29.
- [32] S. Eilenberg, S. MacLane, General theory of natural equivalences, *Transactions of the American Mathematical Society* 58 (1945) 231–294.
- [33] S. Mac Lane, *Categories for the Working Mathematician*, second ed., Springer, 1998.
- [34] M. Barr, C. Wells, *Category Theory for Computing Science*, third ed., Centre de Recherches Mathématiques, Université de Montreal, 1999.
- [35] J. Baez, M. Stay, Physics, topology, logic and computation: A Rosetta Stone, in: B. Coecke (Ed.), *New Structures for Physics*, volume 813 of *Lecture Notes in Physics*, Springer, 2011, pp. 95–172.

- [36] R. Rosen, The representation of biological systems from the standpoint of the theory of categories, *The Bulletin of Mathematical Biophysics* 20 (1958) 317–341.
- [37] G. S. Halford, W. H. Wilson, A category theory approach to cognitive development, *Cognitive Psychology* 12 (1980) 356–411.
- [38] F. Magnan, G. E. Reyes, Category theory as a conceptual tool in the study of cognition., in: J. Macnamara, G. E. Reyes (Eds.), *The Logical Foundations of Cognition*, volume 4 of *Vancouver Studies in Cognitive Science*, Oxford University Press, 1994, pp. 57–90.
- [39] S. Phillips, W. H. Wilson, G. S. Halford, What do transitive inference and class inclusion have in common? Categorical (co)products and cognitive development, *PLoS Computational Biology* 5 (2009) e1000599.
- [40] J. A. Fodor, Z. W. Pylyshyn, Connectionism and cognitive architecture: A critical analysis, *Cognition* 28 (1988) 3–71.
- [41] S. Phillips, W. H. Wilson, Categorical compositionality: A category theory explanation for the systematicity of human cognition, *PLoS Computational Biology* 6 (2010) e1000858.
- [42] S. Phillips, W. H. Wilson, Categorical compositionality II: Universal constructions and a general theory of (quasi-)systematicity in human cognition., *PLoS Computational Biology* 7 (2011) e1002102.
- [43] S. Phillips, W. H. Wilson, Categorical compositionality III: F-(co)algebras and the systematicity of recursive capacities in human cognition., *PLoS ONE* 7 (2012) e35028.
- [44] D. Gentner, Structure-mapping: A theoretical framework for analogy, *Cognitive Science* 7 (1983) 155–170.
- [45] S. Phillips, Analogy, cognitive architecture and universal construction: A tale of two systematicities, *PLoS ONE* 9 (2014) e89152.
- [46] S. Phillips, W. H. Wilson, Second-order systematicity of associative learning: A paradox for classical compositionality and a coalgebraic resolution, *PLoS ONE* 11 (2016) e0160619.
- [47] Z. Arzi-Gonczarowski, Perceive this as that – analogies, artificial perception, and category theory, *Annals of Mathematics and Artificial Intelligence* 26 (1999) 215–252.
- [48] Z. Arzi-Gonczarowski, D. Lehmann, From environments to representations—a mathematical theory of artificial perceptions, *Artificial Intelligence* 102 (1998) 187–247.

- [49] Z. Arzi-Gonczarowski, D. Lehmann, Introducing the mathematical category of artificial perceptions, *Annals of Mathematics and Artificial Intelligence* 23 (1998) 267–298.
- [50] J. A. Navarrete, P. Dartnell, Towards a category theory approach to analogy: Analyzing re-representation and acquisition of numerical knowledge, *PLoS Computational Biology* 13 (2017) e1005683.
- [51] M. J. Healy, T. P. Caudell, *Neural Networks, Knowledge and Cognition: A Mathematical Semantic Model Based upon Category Theory*, UNM Technical Report EECE-TR-04-020, University of New Mexico, 2004.
- [52] M. J. Healy, T. P. Caudell, T. E. Goldsmith, A Model of Human Categorization and Similarity Based Upon Category Theory, UNM Technical Report EECE-TR-08-0010, University of New Mexico, 2008.
- [53] M. J. Healy, T. P. Caudell, Episodic Memory via Spans and Cospans: A Hierarchy of Spatiotemporal Colimits, UNM Technical Report ECE-TR-17-0001, University of New Mexico, 2017.
- [54] J. Gómez-Ramirez, R. Sanz, Hippocampal categories: A mathematical foundation for navigation and memory, in: C. Hernández, R. Sanz, J. Gómez-Ramirez, L. S. Smith, A. Hussain, A. Chella, I. Aleksander (Eds.), *From Brains to Systems. Brain-Inspired Cognitive Systems 2010*, volume 718 of *Advances in Experimental Medicine and Biology*, Springer, 2011, pp. 149–164.
- [55] J. Gómez-Ramirez, A New Foundation for Representation in Cognitive and Brain Science, volume 7 of *Springer Series in Cognitive and Neural Systems*, Springer, 2016.
- [56] D. Ellerman, Adjoints and emergence: applications of a new theory of adjoint functors, *Axiomathes* 17 (2007) 19–39.
- [57] D. Ellerman, Determination through universals: An application of category theory in the life sciences, 2013. ArXiv:1305.6958.
- [58] D. Ellerman, On adjoint and brain functors, *Axiomathes* 26 (2016) 41–61.
- [59] A. Ehresmann, J.-P. Vanbremeersch, Memory Evolutive Systems: Hierarchy, Emergence, Cognition, volume 4 of *Studies in Multidisciplinarity*, Elsevier, 2007.
- [60] A. Ehresmann, J. Gomez-Ramirez, Conciliating neuroscience and phenomenology via category theory, *Progress in Biophysics and Molecular Biology* 119 (2015) 347–359.
- [61] A. Koestler, *The Act of Creation*, Hutchinson & Co., 1964.
- [62] K. Duncker, On Problem-Solving, volume 58 of *Psychological Monographs*, The American Psychological Association, 1945.

- [63] M. Gardner, Mathematical games, *Scientific American* 204 (1961) 166–176.
- [64] J. Goguen, A categorical manifesto, *Mathematical Structures in Computer Science* 1 (1991).
- [65] B. Pierce, *Basic Category Theory for Computer Scientists*, MIT Press, 1991.
- [66] C. B. Jay, Partial functions, ordered categories, limits and cartesian closure, in: G. Birtwistle (Ed.), *IV Higher Order Workshop, Banff 1990*, Springer, 1991, pp. 151–161.
- [67] C. B. Jay, Extending properties to categories of partial maps, LFCS Report ECS-LFCS-90-107, Laboratory for Foundations of Computer Science, University of Edinburgh, 1990.
- [68] W. Kahl, *Collagory Notes, Version 1*, SQRL Report 57, Software Quality Research Laboratory, McMaster University, 2010.
- [69] C. B. Jay, Partial functions, ordered categories, limits and cartesian closure, Unpublished manuscript available at <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.6433&rep=rep1&type=pdf>, 1993. Last accessed on Dec 12, 2019.
- [70] R. Diaconescu, $\frac{3}{2}$ -Institutions: an institution theory for conceptual blending, 2017. ArXiv:1708.09675 [math.LO].
- [71] E. Robinson, G. Rosolini, Categories of partial maps, *Information and Computation* 79 (1988) 95–130.
- [72] B. Mitchell, *Theory of Categories*, Academic Press, 1965.
- [73] O. Kutz, T. Mossakowski, J. Hois, J. Bateman, Ontological blending in DOL, in: T. R. Besold, K.-U. Kühnberger, M. Schorlemmer, A. Smaill (Eds.), *Proceedings of the Workshop “Computational Creativity, Concept Invention, and General Intelligence”*, volume 1-2012 of *PICS Publications of the Institute of Cognitive Science*, Universität Osnabrück, 2012, pp. 33–40.
- [74] B. Carpenter, *The Logic of Typed Feature Structures*, Cambridge University Press, 1992.
- [75] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: L. P. Kaelbling, A. Saffiotti (Eds.), *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, UK, July 30 - August 5, 2005, Professional Book Center, 2005, pp. 364–369.

- [76] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. D. Mosses, D. Sannella, A. Tarlecki, CASL: the common algebraic specification language, *Theoretical Computer Science* 286 (2002) 153–196.
- [77] S. Ontañón, P. Meseguer, Speeding up operations on feature terms using constraint programming and variable symmetry, *Artificial Intelligence* 220 (2015) 104–120.
- [78] A. A. Copestake, Implementing Typed Feature Structure Grammars, volume 110 of *CSLI Lecture Notes*, CSLI Publications, 2002.
- [79] H. Aït-Kaci, A. Podelski, Towards a meaning of LIFE, *Journal of Logic Programming* 16 (1993) 195–234.
- [80] G. Erbach, ProFIT: Prolog with features, inheritance and templates, in: S. P. Abney, E. W. Hinrichs (Eds.), *EACL 1995, 7th Conference of the European Chapter of the Association for Computational Linguistics, March 27-31, 1995, University College Dublin, Belfield, Dublin, Ireland, The Association for Computer Linguistics, 1995*, pp. 180–187.
- [81] H. Aït-Kaci, Data models as constraint systems: A key to the semantic web, *Constraint Programming Letters* 1 (2007) 33–88.
- [82] E. Plaza, Cases as terms: A feature term approach to the structured representation of cases, in: M. M. Veloso, A. Aamodt (Eds.), *Case-Based Reasoning Research and Development, First International Conference, ICCBR-95, Sesimbra, Portugal, October 23-26, 1995, Proceedings, volume 1010 of Lecture Notes in Computer Science*, Springer, 1995, pp. 265–276.
- [83] S. Ontañón, E. Plaza, Similarity measures over refinement graphs, *Machine Learning* 87 (2012) 57–92.
- [84] K. Hirata, T. Aoyagi, Computational music representation based on the generative theory of tonal music and the deductive object-oriented database, *Computer Music Journal* 27 (2003) 73–89.
- [85] G. Gazdar, C. Mellish, *Natural Language Processing in PROLOG*, Addison-Wesley, 1989.
- [86] R. Confalonieri, M. Schorlemmer, O. Kutz, R. Peñaloza, E. Plaza, M. Eppe, Conceptual blending in $\mathcal{EL}++$, in: M. Lenzerini, R. Peñaloza (Eds.), *Proceedings of the 29th International Workshop on Description Logics (DL 2016)*. Cape Town, South Africa, April 22-25, 2016, volume 1577 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016.
- [87] R. Confalonieri, M. Eppe, M. Schorlemmer, O. Kutz, R. Peñaloza, E. Plaza, Upward refinement operators for conceptual blending in the description logic $\mathcal{EL}++$, *Annals of Mathematics and Artificial Intelligence* 82 (2018) 69–99.

- [88] R. Peñaloza, A. Turhan, A practical approach for computing generalization inferences in \mathcal{EL} , in: G. Antoniou, M. Grobelnik, E. P. B. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, J. Z. Pan (Eds.), *The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*, volume 6643 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 410–423.
- [89] A. Ecke, A. Turhan, Role-depth bounded least common subsumers for \mathcal{EL}^+ and \mathcal{ELI} , in: T. Eiter, B. Glimm, Y. Kazakov, M. Krötzsch (Eds.), *Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*, volume 846 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012.
- [90] A. Ecke, R. Peñaloza, A. Turhan, Computing role-depth bounded generalizations in the description logic \mathcal{ELOR} , in: I. J. Timm, M. Thimm (Eds.), *KI 2013: Advances in Artificial Intelligence - 36th Annual German Conference on AI, Koblenz, Germany, September 16-20, 2013. Proceedings*, volume 8077 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 49–60.
- [91] R. Confalonieri, E. Plaza, M. Schorlemmer, A process model for concept invention, in: F. Pachet, A. Cardoso, V. Corruble, F. Ghedini (Eds.), *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*, Sony CSL, Paris, France, 2016, pp. 338–345.
- [92] M. Eppe, R. Confalonieri, E. Maclean, M. Kaliakatsos, E. Cambouropoulos, M. Schorlemmer, K.-U. Kühnberger, Computational invention of cadences and chord progressions by conceptual chord-blending, in: Q. Yang, M. Wooldridge (Eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, Buenos Aires, Argentina, 25–31 July 2015, AAAI Press, 2015, pp. 2445–2451.
- [93] M. Eppe, E. Maclean, R. Confalonieri, M. Schorlemmer, O. Kutz, E. Plaza, ASP, amalgamation, and the conceptual blending workflow, in: F. Calimeri, G. Ianni, M. Truszczynski (Eds.), *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *Lecture Notes in Artificial Intelligence*, Springer, 2015, pp. 309–316.
- [94] M. Eppe, E. Maclean, R. Confalonieri, O. Kutz, M. Schorlemmer, E. Plaza, K.-U. Kühnberger, A computational framework for conceptual blending, *Artificial Intelligence* 256 (2018) 105–129.
- [95] T. Veale, From conceptual “mash-ups” to “bad-ass” blends: A robust computational model of conceptual blending, in: M. L. Maher, K. Hammond, A. Pease, R. Pérez y Pérez, D. Ventura, G. Wiggins (Eds.), *Proceedings of the Third International Conference on Computational Creativity*, 2012, pp. 1–8.

- [96] L. Brandt, P. A. Brandt, Making sense of a blend: A cognitive-semiotic approach to metaphor, *Annual Review of Cognitive Linguistics* 3 (2005) 216–249.
- [97] J. A. Goguen, D. F. Harrell, Foundations for active multimedia narrative: Semiotic spaces and structural blending, Unpublished manuscript available at <https://cseweb.ucsd.edu/~goguen/pps/narr.pdf>, n.d. Last accessed on Dec 12, 2019.
- [98] D. F. Harrell, Theory and Technology for Computational Narrative: An Approach to Generative and Interactive Narrative with Bases in Algebraic Semiotics and Cognitive Linguistics, Ph.D. thesis, University of California, San Diego, 2007.
- [99] J. A. Goguen, K. Lin, G. Roşu, Circular coinductive rewriting, in: The Fifteenth IEEE International Conference on Automated Software Engineering, ASE 2000, Grenoble, France, September 11-15, 2000, IEEE Computer Society, 2000, pp. 123–132.
- [100] F. C. Pereira, A. Cardoso, Optimality principles for conceptual blending: A first computational approach, *AISB Journal* 1 (2003).
- [101] F. C. Pereira, A. Cardoso, Experiments with free concept generation in Divago, *Knowledge-Based Systems* 19 (2006) 459–470.
- [102] P. Martins, H. G. Oliveira, J. ao Carlos Gonçalves, M. v. António Cruz, F. Amílcar Cardoso, N. Lavrač, S. Linkola, H. Toivonen, R. Hervá, G. Méndez, P. Gervás, Computational creativity infrastructure for on-line software composition: A conceptual blending use case, *IBM Journal of Research & Development* 63 (2019) 9:1–9:17.
- [103] M. Guhe, A. Pease, A. Smaill, M. Martínez, M. Schmidt, H. Gust, K.-U. Kühnberger, U. Krumnack, A computational account of conceptual blending in basic mathematics, *Cognitive Systems Research* 12 (2011) 249–265.
- [104] A. Schwering, U. Krumnack, K.-U. Kühnberger, H. Gus, Syntactic principles of heuristic-driven theory projection, *Cognitive Systems Research* 10 (2009) 251–269.
- [105] M. Martínez, A. M. H. Abdel-Fattah, U. Krumnack, D. Gómez-Ramírez, A. Smaill, T. R. Besold, A. Pease, M. Schmidt, M. Guhe, K.-U. Kühnberger, Theory blending: extended algorithmic aspects and examples, *Annals of Mathematics and Artificial Intelligence* 80 (2017) 65–89.
- [106] R. Confalonieri, A. Pease, M. Schorlemmer, T. R. Besold, O. Kutz, E. Maclean, M. Kaliakatsos-Papakostas (Eds.), *Concept Invention: Foundations, Implementation, Social Aspects and Applications*, Computational Synthesis and Creative Systems, Springer, 2018.

- [107] T. Mossakowski, M. Codescu, F. Neuhaus, O. Kutz, The distributed ontology, modeling and specification language — DOL, in: A. Koslow, A. Buchsbaum (Eds.), *The Road to Universal Logic. Festschrift for the 50th Birthday of Jean-Yves Béziau*, volume II of *Studies in Universal Logic*, Birkhäuser, 2015, pp. 489–520.
- [108] M. Schorlemmer, R. Confalonieri, E. Plaza, The Yoneda path to the Buddhist monk blend, in: O. Kutz, S. de Cesare, M. M. Hedblom, T. R. Besold, T. Veale, F. Gailly, G. Guizzardi, M. Lycett, C. Partridge, O. Pastor, M. Grüninger, F. Neuhaus, T. Mossakowski, S. Borgo, L. Bozzato, C. D. Vescovo, M. Homola, F. Loebe, A. Barton, J. Bourguet (Eds.), *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., volume 1660 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016.
- [109] M. Johnson, *The Body in the Mind*, The University of Chicago Press, 1987.
- [110] G. Lakoff, *Women, Fire and Dangerous Things*, University Of Chicago Press, 1987.
- [111] G. Mazzola, J. Park, F. Thalmann, *Musical Creativity*, Computational Music Science, Springer, 2011.
- [112] M. Andreatta, A. Ehresmann, R. Guitart, G. Mazzola, Towards a categorical theory of creativity for music, discourse, and cognition, in: J. Yust, J. Wild, J. A. Burgoyne (Eds.), *Mathematics and Computation in Music. 4th International Conference, MCM 2013. Montreal, QC, June 2013. Proceedings*, volume 7937 of *Lecture Notes in Artificial Intelligence*, Springer, 2013, pp. 19–37.
- [113] G. Lakoff, M. Johnson, *Philosophy in the Flesh*, Basic Books, 1999.
- [114] M. M. Hedblom, O. Kutz, R. Peñaloza, G. Guizzardi, Image schema combinations and complex events, *Künstliche Intelligenz* 33 (2019) 279–291.