
Analogy, Amalgams, and Concept Blending

Tarek R. Besold

TBESOLD@UNI-OSNABRUECK.DE

Kai-Uwe Kühnberger

KKUEHNBE@UNI-OSNABRUECK.DE

Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany

Enric Plaza

ENRIC@IIIA.CSIC.ES

IIIA, Artificial Intelligence Research Institute, CSIC, Spanish Council for Scientific Research, Barcelona, Catalonia (Spain)

Abstract

Concept blending — a cognitive process which allows for the combination of certain elements (and their relations) from originally distinct conceptual spaces into a new unified space combining these previously separate elements, and enables reasoning and inference over the combination — is taken as a key element of creative thought and combinatorial creativity. In this paper, we provide an intermediate report on work towards the development of a computational-level and algorithmic-level account of concept blending. We present the theoretical background as well as an algorithmic proposal combining techniques from computational analogy-making and case-based reasoning, and exemplify the feasibility of the approach in two case studies.

1. Introduction: Computational Creativity and Concept Blending

Boden (2003) identifies three forms of creativity: exploratory, transformational, and combinatorial. The label exploratory refers to creativity which arises from a thorough and persistent search of a well-understood domain (i.e., within an already established conceptual space), whilst transformational creativity either involves the removal of constraints and limitations from the initial domain definition, or the rejection of characteristic assumptions forming part of the specification of the creative problem (or both). Combinatorial creativity shares traits of both other forms in that it arises from a combinatorial process joining familiar ideas (in the form of, for instance, concepts, theories, or artworks) in an unfamiliar way, by this producing novel ideas.

Computationally modeling the latter form of creativity turns out to be surprisingly complicated: although the overall idea of combining preexisting ideas into new ones seems fairly intuitive and straightforward, when looking at it from a more formal perspective at the current stage neither can a precise algorithmic characterization be given, nor are at least the details of a possible computational-level theory describing the process(es) at work well understood. Still, in recent years a proposal by Fauconnier and Turner (1998) called concept blending (or conceptual integration) has influenced and reinvigorated studies trying to unravel the general cognitive principles operating during creative thought. In their theory, concept blending constitutes a cognitive process which allows for the combination of certain elements (and their relations) from originally distinct conceptual spaces into

a new unified space combining these previously separate elements, and then enables reasoning and inference over the combination.

Unfortunately, Fauconnier and Turner (1998) (also in their later works) remain mostly silent concerning details needed for a proper computational modeling of concept blending as cognitive capacity — neither do they provide a fully worked out and formalized theory themselves, nor does their informal account capture key properties and functionalities as, for example, the retrieval of input spaces, the selection and transfer of elements from the input into the blend space, or the further combination of possibly mutually contradictory elements in the blend. In short: Up until today, the theory does not specify how the blending process is supposed to work.

These shortcomings notwithstanding, several researchers in AI and computational cognitive modeling have used the provided conceptual descriptions as starting point for suggesting possible refinements and implementations: (Goguen & Harrell, 2010) propose a concept blending-based approach to the analysis of the style of multimedia content in terms of blending principles and also provide an experimental implementation, (Pereira, 2007) tries to develop a computationally plausible model of several hypothesized sub-parts of concept blending, (Thagard & Stewart, 2011) exemplify how creative thinking could arise from using convolution to combine neural patterns into ones which are potentially novel and useful, and (Veale & O’Donoghue, 2000) present their computational model of conceptual integration and propose several extensions to the at the time actual view on concept blending.

Since 2013, another attempt at developing a computationally feasible, cognitively-inspired formal model of concept creation, grounded on a sound mathematical theory of concepts and implemented in a generic, creative computational system is undertaken by a European research consortium in the so called Concept Invention Theory (COINVENT) project.¹ One of the main goals of the COINVENT research program is the development of a computational-level and algorithmic-level account of concept blending based on insights from psychology, AI, and cognitive modeling, the heart of which are made up by results from cognitive systems studies on computational analogy-making and knowledge transfer and combination (i.e., the computation of so called “amalgams”) from case-based reasoning. In the following we present an analogy-inspired perspective on the COINVENT core model for concept blending and show how the respective mechanisms and systems interact.

2. Computational Models of Analogy and Amalgams

As analogy seems to play a crucial role in human cognition (Gentner & Smith, 2013), researchers on the computational side of cognitive science and in AI also very quickly got interested in the topic and have been creating computational models of analogy-making basically since the advent of computer systems, among others giving rise to Winston (1980)’s work on analogy and learning, Hofstadter and Mitchell (1994)’s Copycat system, or Falkenhainer et al. (1989)’s well-known Structure-Mapping Engine. One of the latest entries in the long series of computational analogy engines, and the system applied in COINVENT, is the Heuristic-Driven Theory Projection (HDTP) framework (Schmidt et al., 2014), a generalization-based symbolic analogy engine discussed in detail in Sect. 2.1.

1. Also see <http://www.coinvent-project.eu> for details on the consortium and the project.

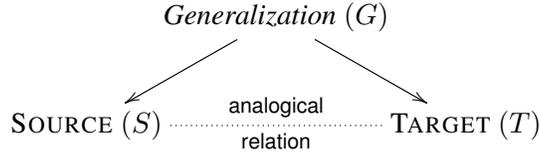


Figure 1. A schematic overview of HDTP’s generalization-based approach to analogy.

In a conceptually related, but mostly independently conducted line of work researchers in case-based reasoning (CBR) have been trying to develop problem solving methodologies based on the principle that similar problems likely tend to have similar solutions. CBR tries to solve problems by retrieving one or several relevant cases for the current issue at hand from a case-base with already solved previous problems, and then reusing the knowledge to also tackle the new task (Aamodt & Plaza, 1994). While the retrieval stage has received significant attention over the last two decades, the transfer and combination of knowledge from the retrieved case to the current problem has been studied only to a lesser extent, with (Ontañón & Plaza, 2012) being a recent attempt at also gaining insights on this part of the CBR cycle by suggesting the framework of amalgams (Ontañón & Plaza, 2010) as a formal model. Sect. 2.2 gives an overview of amalgams as used in COINVENT.

2.1 Generalization-Based Analogy-Making Using Heuristic-Driven Theory Projection

Heuristic-Driven Theory Projection has been conceived as a mathematically sound theoretical model and implemented engine for computational analogy-making, computing analogical relations and inferences for domains which are presented in (possibly different) many-sorted first-order logic languages: Source and target of the analogy-making process are defined in terms of axiomatizations, i.e., given by a finite set of formulae. HDTP follows a generalization-based approach to analogy-making: Given both domains, a common generalization encompassing structurally shared elements common to both input domains is computed (*mapping phase*) and this generalization then guides the analogical alignment and knowledge transfer process of unmatched knowledge from the source to the target domain used for establishing new hypotheses (*transfer phase*). See Fig. 1 for a conceptual overview of the entire analogy mechanism.

More precisely, HDTP uses many-sorted term algebras to define the input conceptual domains (i.e., one source and one target domain for the later analogy). A term algebra requires two ingredients: a *signature* and a set of *variables*.

Definition 1 A many-sorted signature $\Sigma = \langle \text{Sort}, \text{Func} \rangle$ is a tuple containing a finite set *Sort* of sorts, and a finite set *Func* of function symbols. An n -ary function symbol $f \in \text{Func}$ is specified by $f : s_1 \times s_2 \times \dots \times s_n \rightarrow s$, where $s, s_1, \dots, s_n \in \text{Sort}$. We will consider function symbols of any non-negative arity, and we will use 0-ary function symbols to represent constants.

Definition 2 Let $\Sigma = \langle \text{Sort}, \text{Func} \rangle$ be a many-sorted signature, and let $\mathcal{V} = \{x_1 : s_1, x_2 : s_2, \dots\}$ be an infinite set of sorted variables, where the sorts are chosen from *Sort*. Associated with each variable $x_i : s_i$ is an arity, analogous to the arity of function symbols above. For any $i \geq 0$, we let

\mathcal{V}_i be the variables of arity i . The set $Term(\Sigma, \mathcal{V})$ and the function $sort : Term(\Sigma, \mathcal{V}) \rightarrow Sort$ are defined inductively as follows:

1. If $x : s \in \mathcal{V}$, then $x \in Term(\Sigma, \mathcal{V})$ and $sort(x) = s$.
2. If $f : s_1 \times s_2 \times \dots \times s_n \rightarrow s$ is a function symbol in Σ , and $t_1, \dots, t_n \in Term(\Sigma, \mathcal{V})$ with $sort(t_i) = s_i$ for each i , then $f(t_1, \dots, t_n) \in Term(\Sigma, \mathcal{V})$ with $sort(f(t_1, \dots, t_n)) = s$.

We refer to the structure $\langle Term(\Sigma, \mathcal{V}), sort \rangle$ as a term algebra, often suppressing $sort$.

As an example for a domain representation using HDTP’s language, Table 1 reproduces a possible formalization of the concept of “horse” using some of a horse’s key characteristics (this formalization reoccurs below as part of a bigger example demonstrating the conceptual blending capacities of the framework combining analogy and amalgams in Sect. 3.4):

Table 1. Example formalization of a stereotypical characterization of a horse.

Sorts:			
clade, entity, bodypart, ability			
Entities:			
mammal	: clade	horse	: entity torso, legs
			: bodypart walk
			: ability
Predicates:			
is_of_clade	: entity \times clade,	has_bodypart	: entity \times bodypart,
		has_ability	: entity \times ability
Laws of the horse characterization:			
is_of_clade	(horse, mammal)	has_bodypart	(horse, legs)
		has_ability	(horse, walk)

Given two input domains, HDTP uses anti-unification to compute a generalization of both domains. In this process, terms are generalized resulting in an anti-instance, where differing subterms are replaced by variables; the original terms can be restored by inverting the procedure, i.e., by replacing the new variables by appropriate subterms (also see Fig. 2 below). These “replacements” can be formalized as substitutions:

Definition 3 Given term algebra $Term(\Sigma, \mathcal{V})$. A term substitution is a partial function $\sigma : \mathcal{V} \rightarrow Term(\Sigma, \mathcal{V})$ mapping variables to terms, formally represented by $\sigma = \{x_1 \rightarrow t_1, \dots, x_n \rightarrow t_n\}$ provided each of the x_i is unique and the sorts of the variables and terms match. An application of a substitution σ on a term is defined inductively by:

1. $apply(x, \sigma) = \begin{cases} t & x \rightarrow t \in \sigma \\ x & \text{otherwise.} \end{cases}$
2. $apply(f(t_1, \dots, t_n), \sigma) = f(apply(t_1, \sigma), \dots, apply(t_n, \sigma))$.

Given two terms t, t' and a substitution σ such that $apply(t, \sigma) = t'$, then we call t' an instance of t and t an anti-instance of t' . We will often shorten $apply(t, \sigma) = t'$ to $t \xrightarrow{\sigma} t'$, or $t \rightarrow t'$ if the substitution is clear from context.

Using substitutions, also generalizations can formally be characterized, with the *least general generalization* playing a special role as most specific anti-unifier (i.e., as minimal with respect to the instantiation order):

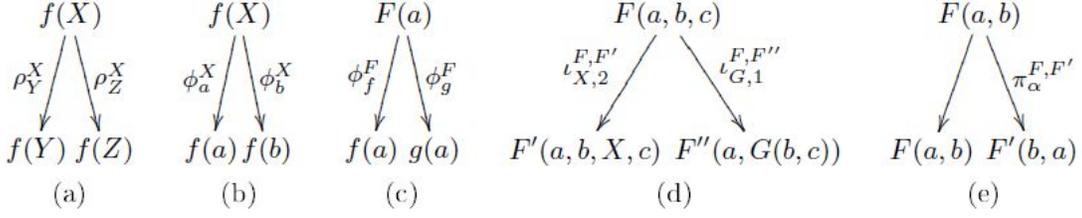


Figure 2. Examples for all four types of unit substitutions admissible in HDTP's restricted form of higher-order anti-unification: (a) shows the variable renamings X/Y and X/Z , (b) exemplifies fixations of argument variables and (c) gives the corresponding counterparts for function variables, (d) represents two argument insertions, and (e) illustrates an argument permutation.

Definition 4 Let f, g be terms from a term algebra $Term(\Sigma, \mathcal{V})$. A generalization of f and g is a triple $\langle G, \sigma, \tau \rangle$ where $G \in Term(\Sigma, \mathcal{V})$ and σ, τ are substitutions such that $G \xrightarrow{\sigma} f$ and $G \xrightarrow{\tau} g$. The generalization $\langle G, \sigma, \tau \rangle$ is called the least general generalization (LGG) if for any generalization $\langle G', \sigma', \tau' \rangle$ of f, g , there exists a substitution ϕ such that $G' \xrightarrow{\phi} G$.

As shown by Plotkin (1970), the LGG is *unique* when considering only first-order anti-unification between terms.

Against this background, Schwering et al. (2009) describe a restricted form of higher-order anti-unification applied in HDTP, defined as using the composition of a number of unit substitutions operating on higher-order terms.

Definition 5 The following are the types of unit substitutions allowed in restricted higher-order anti-unification.

1. A renaming $\rho(F, F')$ replaces a variable $F \in \mathcal{V}_n$ with another variable $F' \in \mathcal{V}_n$:

$$F(t_1, \dots, t_n) \xrightarrow{\rho(F, F')} F'(t_1, \dots, t_n).$$
2. A fixation $\phi(F, f)$ replaces a variable $F \in \mathcal{V}_n$ with a function symbol $f \in \mathcal{C}_n$:

$$F(t_1, \dots, t_n) \xrightarrow{\phi(F, f)} f(t_1, \dots, t_n).$$
3. An argument insertion $\iota(F, F', V, i)$ is defined as follows, for $F \in \mathcal{V}_n, F' \in \mathcal{V}_{n-k+1}, V \in \mathcal{V}_k, i \in [n]$:

$$F(t_1, \dots, t_n) \xrightarrow{\iota(F, F', V, i)} F'(t_1, \dots, t_{i-1}, V(t_i, \dots, t_{i+k}), t_{i+k+1}, \dots, t_n).$$
 It “wraps” k of the subterms in a term using a k -ary variable, or can be used to insert a 0-ary variable.
4. A permutation $\pi(F, \tau)$ rearranges the arguments of a term, with $F \in \mathcal{V}_n, \tau : [n] \rightarrow [n]$ a bijection:

$$F(t_1, \dots, t_n) \xrightarrow{\pi(F, \tau)} F(t_{\pi(1)}, \dots, t_{\pi(n)}).$$

A restricted substitution is a substitution $t \rightarrow t'$ which results from the composition of any sequence of unit substitutions transforming t into t' .

Fig. 2 gives generic examples for the different types of unit substitutions in HDTP's version of restricted higher-order anti-unification.

Clearly, restricted substitutions are strictly more general than mere (first-order) term substitutions. While for a given term t there are (up to renaming) still only finitely many anti-instances (i.e.,

terms s with $s \rightarrow t$), this generality unfortunately, among others, causes the LGG to be no longer necessarily unique. Therefore, HDTP ranks generalizations according to a complexity order on the complexity of generalization (based on a complexity measure for substitutions), and finally chooses the least complex generalizations as preferred ones.

From a practical point of view, it is also necessary to anti-unify not only terms, but formulae: HDTP extends the notion of generalization also to formulae by basically treating formulae in clause form and terms alike (as positive literals are structurally equal to function expressions, and complex clauses in normal form may be treated component wise).

Furthermore, analogies in general not only rely on an isolated pair of formulae from source and target, but on two sets of formulae, making it necessary to extend the notion of anti-unification accordingly:

Definition 6 Let $Th(Ax)$ denote the set of all formulas that can be syntactically derived from a set of axioms Ax , i.e., $Th(Ax) = \{\phi \mid Ax \vdash \phi\}$, and let G be a finite set of formulae.

G is an anti-instance of a set of formulae F if and only if there exists a substitution σ such that $Th(\text{apply}(G, \sigma)) \subseteq Th(F)$. Given substitutions σ and τ , $\langle G, \sigma, \tau \rangle$ is a generalization of two sets of formulae S and T if and only if $G \xrightarrow{\sigma} S$ and $G \xrightarrow{\tau} T$.

As a simple example, we take the “horse” formalization from Table 1 and the stereotypical characterization of a dog given in Table 2, and generalize them into the shared generalization in Table 3.

Table 2. Example formalization of a stereotypical characterization of a dog.

Sorts:
clade, entity, bodypart, ability

Entities:
mammal : clade dog : entity tail : bodypart drool : ability

Predicates:
is_of_clade : entity \times clade, has_bodypart : entity \times bodypart, has_ability : entity \times ability

Laws of the dog characterization:
*is_of_clade(dog, mammal) has_bodypart(dog, tail)
 has_ability(dog, drool)*

Table 3. Shared generalization of the “horse” and “dog” formalizations from Tables 1 and 2, respectively.

Sorts:
clade, entity, bodypart, ability

Entities:
mammal : clade E : entity B : bodypart A : ability

Predicates:
is_of_clade : entity \times clade, has_bodypart : entity \times bodypart, has_ability : entity \times ability

Laws of the horse characterization:
is_of_clade(E, mammal) has_bodypart(E, B) has_ability(E, A)

When processing sets of formulae, a heuristic is applied for iteratively selecting pairs of formulae to be generalized: Coherent mappings outmatch incoherent ones, i.e., mappings in which substitutions can be reused are preferred over isolated substitutions, as they are assumed to be better suited to induce the analogical relation.

Finally, HDTP in its heuristics also aims to maximize the coverage of generalizations:

Definition 7 Given a generalization $\langle G, \sigma, \tau \rangle$ of two sets of axioms S and T , $Th(\text{apply}(G, \sigma)) \subseteq Th(S)$ is said to be covered by G , and for T accordingly.

The degree to which G covers $Th(S)$ is called the coverage of G , with a generalization $\langle G, \sigma, \tau \rangle$ having at least the same coverage as $\langle G', \sigma', \tau' \rangle$ if there exists a substitution θ for which it holds that $G' \xrightarrow{\theta} G$, $\sigma' = \sigma \circ \theta$, and $\tau' = \tau \circ \theta$ (inducing a partial order over generalizations).

In general, while there are some constraining factors which have to be taken into account (Schwering et al., 2009), maximizing the coverage of an analogy seems meaningful as this also automatically increases the domain support for the corresponding analogy.

Once obtained, the generalized theory and the substitutions specify the analogical relation, and formulae of the source for which no correspondence in the target domain can be found may, by means of the already established substitutions, be transferred to the target, constituting a process of analogical transfer between the domains.

Concerning applications of HDTP, for instance Guhe et al. (2010) apply the framework in modeling a potential inductive analogy-based process for establishing the fundamental concepts of arithmetics, Schwering et al. (2009) use HDTP for solving geometric analogies, and Besold (2014) summarizes a series of studies applying the framework to modeling analogy use in education and teaching situations.

2.2 Combining Conceptual Theories Using Amalgams

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

Formally, the notion of amalgams can be defined in any representation language \mathcal{L} for which a subsumption relation \sqsubseteq between the formulas (or descriptions) of \mathcal{L} can be defined. We say that a description I_1 subsumes another description I_2 ($I_1 \sqsubseteq I_2$) when I_1 is more general (or equal) than I_2 . Additionally, we assume that \mathcal{L} contains the infimum element \perp (or ‘any’), and the supremum element \top (or ‘none’) with respect to the subsumption order.

Next, for any two descriptions I_1 and I_2 in \mathcal{L} we can define their *unification*, $(I_1 \sqcup I_2)$, which is the *most general specialization* of two given descriptions, and their *anti-unification*, $(I_1 \sqcap I_2)$, defined as the *least general generalization* of two descriptions, representing the most specific description that subsumes both. Intuitively, a unifier is a description that has all the information in both the original descriptions; when joining this information yields to inconsistency this is equivalent to say that $I_1 \sqcup I_2 = \top$, they have no common specialization except ‘none’. The anti-unification $I_1 \sqcap I_2$ contains all that is common to both I_1 and I_2 ; when they have nothing in common then $I_1 \sqcap I_2 = \perp$. Depending on \mathcal{L} anti-unification and unification might be unique or not.

The notion of *amalgam* can be conceived of as a generalization of the notion of unification: as ‘partial unification’ (Ontañón & Plaza, 2010). Unification means that what is true for I_1 or I_2 is also

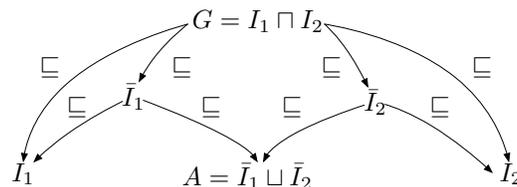


Figure 3. A diagram of an amalgam A from inputs I_1 and I_2 where $A = \bar{I}_1 \sqcup \bar{I}_2$.

true for $I_1 \sqcup I_2$; e.g. if I_1 describes ‘a red vehicle’ and I_2 describes ‘a German minivan’ then their unification yields a common specialization like ‘a red German minivan.’ Two descriptions may possess information that yields an inconsistency when unified; for instance ‘a red French sedan’ and ‘a blue German minivan’ have no common specialization except \top . An *amalgam* of two descriptions is a new description that contains *parts from these two descriptions*. For instance, an amalgam of ‘a red French sedan’ and ‘a blue German minivan’ is ‘a red German sedan’; clearly there are always multiple possibilities for amalgams, like ‘a blue French minivan’.

For the purposes of this paper we can define an amalgam of two input descriptions as follows:

Definition 8 (Amalgam) A description $A \in \mathcal{L}$ is an amalgam of two inputs I_1 and I_2 (with anti-unification $G = I_1 \sqcap I_2$) if there exist two generalizations \bar{I}_1 and \bar{I}_2 such that (1) $G \sqsubseteq \bar{I}_1 \sqsubseteq I_1$, (2) $G \sqsubseteq \bar{I}_2 \sqsubseteq I_2$, and (3) $A = \bar{I}_1 \sqcup \bar{I}_2$

When the \bar{I}_1 and \bar{I}_2 have no common specialization then trivially $A = \top$, since their only unifier is “none”. For our purpose we will be only interested in non-trivial amalgams.

This definition is illustrated in Fig. 3, where the anti-unification of the inputs is indicated as G , and the amalgam A is the unification of two concrete generalizations \bar{I}_1 and \bar{I}_2 of the inputs. Equality here should be understood as \sqsubseteq -equivalence: $X = Y$ iff $X \sqsubseteq Y$ and $Y \sqsubseteq X$. Conventionally, we call the *space of amalgams* of I_1 and I_2 the set of all amalgams A that satisfy the definition above.

Usually we are interested only on maximal amalgams of two input descriptions, i.e., those amalgams that contain maximal parts of their inputs that can be unified into a new coherent description. Formally, an amalgam A of inputs I_1 and I_2 is maximal if there is no other non-trivial amalgam A' of inputs I_1 and I_2 such that $A \sqsubset A'$. The reason why we are interested in maximal amalgams is very simple: a non-maximal amalgam $\bar{A} \sqsubset A$ preserves less compatible information from the inputs than the maximal amalgam A ; conversely, any non-maximal amalgam \bar{A} can be obtained by generalizing a maximal amalgam A , since $\bar{A} \sqsubset A$.

There is a special case of particular interest that is called *asymmetric amalgam*, in which the two inputs play different roles. The inputs are called source and target, and while the source is allowed to be generalized, the target is not.

Definition 9 (Asymmetric Amalgam) An asymmetric amalgam $A \in \mathcal{L}$ of two inputs S (source) and T (target) satisfies that $A = S' \sqcup T$ for some generalization of the source $S' \sqsubseteq S$.

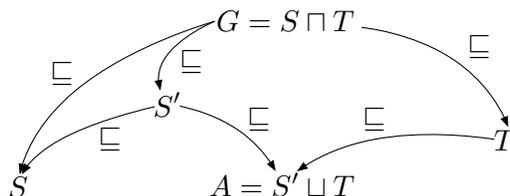


Figure 4. A diagram that transfers content from source S to a target T via an asymmetric amalgam A .

As shown in Fig. 4, the content of target T is transferred completely into the asymmetric amalgam, while the source S is generalized. The result is a form of partial unification that preserves all the information in T while relaxing S by generalization and then unifying one of those generalization S' with T itself. As before, we will usually be interested in maximal amalgams: in this case, a maximal amalgam corresponds to transferring as much content from S to T while keeping the resulting amalgam A consistent. For these reason, asymmetric amalgam can be seen as a model of analogical inference, where information from the source is transferred to the target by creating a new amalgam A that enriches the target T with the content of S' (Ontañón & Plaza, 2012).

3. COINVENT’s Account of Analogy-Based Concept Blending

3.1 The Idea(s) Behind Concept Blending in COINVENT: An Analogy-Inspired View

One of the early formal accounts on concept blending, which is especially influential to the approach applied in COINVENT, is the classical work by Goguen using notions from algebraic specification and category theory (Goguen, 2006). This version of concept blending can be described by the diagram in Fig. 5, where each node stands for a representation an agent has of some concept or conceptual domain. We will call these representations “conceptual spaces” and in some cases abuse terminology by using the word “concept” to really refer to its representation by the agent. The arrows stand for morphisms, that is, functions that preserve at least part of the internal structure of the related conceptual spaces. The idea is that, given two conceptual spaces I_1 and I_2 as input, we look for a generalization G and then construct a blend space B in such a way as to preserve as many as possible structural alignments between I_1 and I_2 established by the generalization. This may involve taking the functions to B to be *partial*, in that not all the structure from I_1 and I_2 might be mapped to B . In any case, as the blend respects (to the largest possible extent) the relationship between I_1 and I_2 , the diagram will commute.

Concept invention by concept blending can then be phrased as the following task: given two axiomatizations of two domain theories I_1 and I_2 , we need first, to compute a generalized theory G of I_1 and I_2 (which codes the commonalities between I_1 and I_2) and second, to compute the blend theory B in a structure preserving way such that new properties hold in B . Ideally, these new properties in B are considered to be (moderately) interesting properties. In what follows, for reasons of simplicity and without loss of generality we assume that the additional properties are just provided by one of the two domains, i.e., we align the situation with a standard setting in computational analogy-making by renaming I_1 and I_2 : The domain providing the additional

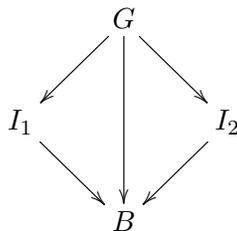


Figure 5. A conceptual overview of (Goguen, 2006)’s account of conceptual blending.

properties for the concept blend will be called source S , the domain providing the conceptual basis and receiving the additional features will be called target T .

The reasoning process in COINVENT is then triggered by the computation of the generalization G (generic space) by HDTP. For concept invention we will only need the mapping mechanism and replace the transfer phase by a new blending algorithm. The mapping is achieved via the usual generalization process between S and T , in which pairs of formulas from the source and target spaces are anti-unified resulting in a generalized theory that reflects common aspects of both spaces. The generalized theory can be projected into the original spaces by substitutions which are computed during anti-unification. In what follows, we will say that a formula is *covered* by the analogy, if it is in the image of this projection (T_c and S_c , respectively), otherwise it is *uncovered* (also see Sect. 2.1). While in analogy making the analogical relations are used in the transfer phase to translate additional uncovered knowledge from the source to the target space, blending combines additional (uncovered) facts from one or both spaces. Therefore the process of blending can build on the generalization and substitutions provided by the analogy engine, but has to include a new mechanism for transfer and concept combination. Here, amalgams naturally come into play: The set of substitutions can be inverted and applied to generalize the original source theory S into a more general version S' (forming a superset of the shared generalization G , also including previously uncovered knowledge from the source) which then can be combined into an asymmetric amalgam with the target theory T , forming the (possibly underspecified) proto-blend T' of both. In a final step, T' is then completed into the blended theory and output of the process T_B by applying corresponding specialization steps stored from the generalization process between S and T (see also Fig. 6).

3.2 Implementing the Model Using HDTP and Asymmetric Amalgams

One of the project aims of COINVENT is an implementation of the general blending model sketched in the previous section into a system producing novel and useful output theories, fully integrating HDTP and the amalgam framework. In what follows, we present an intermediate version on the way to this goal: A blend is taken to be novel if it is not a subset of or equal to the source or the target domain, usefulness is defined as consistency of the resulting theory, the generalization step uses a further constrained variant of restricted higher-order anti-unification, applying only fixations and renamings, the amalgamation uses higher-order unification as combination mechanism, and logical semantic consequence serves as ordering relationship:

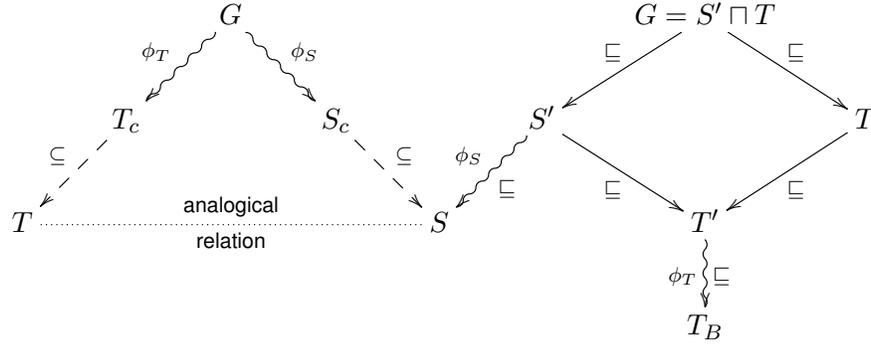


Figure 6. A conceptual overview of the analogy-inspired perspective on COINVENT's account of concept blending as described in Sect. 3.1: The shared generalization G from S and T is computed with $\phi_S(G) = S_c$. The relation ϕ_S is subsequently re-used in the generalization of S into S' , which is then combined in an asymmetric amalgam with T into the proto-blend $T' = S' \sqcup T$ and finally, by application of ϕ_T , completed into the blended output theory T_B . (\subseteq indicates an element-wise subset relationship between sets of axioms and \sqsubseteq indicates subsumption between theories in the direction of the respective arrows.)

1. Given two input domain theories S and T , the (set of) common generalization(s) $G = \{G_1, G_2, \dots, G_n\}$, i.e., the anti-unified forms of sets of axioms which are structurally shared between S and T is computed. In COINVENT, HDTP is used for this step: In the present version of the algorithm, only renamings and fixations are used as admissible types of unit substitutions. As the least general generalization under restricted higher-order anti-unification is (still) not unique (also see Sect. 2.1), the anti-unification itself returns several possible least general generalizations out of which the system chooses one generalization G_x using heuristics.
2. Given this generalized theory G_x , together with the associated two sets of substitutions $\phi_{x,S}$ and $\phi_{x,T}$ respectively corresponding to the covered parts $S_c \subseteq S$ and $T_c \subseteq T$ of the input domain theories, the set of higher-order anti-unifications $\phi_{x,S}^{-1}$ (inversely related to the substitutions $\phi_{x,S}$) is then used to generalize the previous source domain theory S as far as possible into the generalized source theory S' such that $\phi_{x,S}(S') = S$. Here, if $S = S_c$, i.e., all axioms from S could be matched and anti-unified with axioms from T in the previous step (constituting a pathological case as S is supposed to provide some additional content over and above T), it holds that $S = \phi_{x,S}(G_x)$ and, thus, $S' = G_x$; otherwise $G_x \subseteq S'$. Notice that, due to the restriction to fixations and renamings in the higher-order anti-unifications, it holds that $S \models S' \models G$ in both cases. (Here, \subseteq indicates an element-wise subset relationship between sets of axioms as in Fig. 6, and \models indicates the classical semantic consequence relation in the logical sense.)
3. Now, given S' , we can compute the asymmetric amalgam between S' and T (with T staying fixed) using higher-order unification and the semantic consequence relation as subsumption relation for refinement (i.e., given two theories A and B it holds that A is more general than B , $A \sqsubseteq B$, if and only if $B \models A$): Axioms from S' and T are unified pairwise as far as

possible (i.e., a subset of S' of maximum cardinality is unified with a similar subset of T). Conveniently, for the part of S' which is contained in S_c under $\phi_{x,S}$ this is equal to T_c , so only axioms from $\{a \mid a \in S' \wedge \phi_{x,S}(a) \notin S_c\}$ and from $T \setminus T_c$, respectively, have to be checked.² Subsequently, the remaining axioms from both theories are added as additional elements to the resulting unified set of axioms, resulting in an enriched target theory T' .³

4. Remaining variables not instantiated by the unification step between S' and T' (i.e., imported in axioms from S') in the enriched target theory (or proto-blend) T' are instantiated by applying the set of substitutions $\phi_{x,T}$ from the initial generalization step to T' , resulting in the (fully instantiated) blended theory T_B . (If T' does not contain any variables it trivially holds that $T' = \phi_{x,T}(T') = T_B$ and this step becomes obsolete.)
5. A check for consistency of the blended theory T_B is conducted (both internally as well as with respect to potentially available world knowledge). As we are only interested in non-trivial amalgams (i.e., consistent output theories), if an inconsistency is found, clash resolution tries to solve the inconsistency by returning to step 1., removing one or several axioms from S resulting in a new source theory $S_{clash} \subseteq S$, and then re-initiating the procedure.

The resulting blend theory T_B is based on T , (consistently) enriched by imported “unaffected” axioms and (via generalization from S to S' , and re-instantiation from T' to T_B) adapted structural elements from S . This blend forms the (in a certain concept theoretical sense) “closest” blend to T and can presumably play an important role in different contexts: for instance it can account for the addition of new solution elements to a solution idea at hand in problem-solving scenarios, and in creativity tasks the addition of novel features and elements to existing concepts can be achieved.

Clearly, this is only a partial solution on the way to meeting COINVENT’s aims, requiring further algorithmic and conceptual development: The inconsistency resolution in step 5 can probably be made significantly more efficient by developing heuristics for efficiently selecting axioms for removal, the simple identification between usefulness and consistency might not be enough for many contexts, and methods for assessing the novelty of the resulting blend (also allowing for comparisons between different possible blends) have to be developed and integrated.

In the following section, we want to exemplify our approach in two application cases: A concept blending account on Rutherford’s classical analogy between the solar system and the atom, and a re-creation of Pegasus as concept blend between a horse and a bird in classical mythology.

3.3 Example 1: Rutherford’s Model of the Solar System as Scientific Concept Blending

Rutherford’s analogy for deriving his model of the atom from a theory of the solar system, together with certain experimental observations and physical laws known by his time, has become a running example in the literature on computational analogy-making —see, for instance, (Falkenhainer, Forbus, & Gentner, 1989; Schwering et al., 2009). In what follows we will show that it can also be understood as a concept blending process and can be modeled with the approach presented above.

2. The maximality of the outcome is rooted in HDTP’s previously mentioned coverage maximization.

3. Note that the unifications and addition of axioms conserve the \models relation between theories and, thus, the subsumption ordering as indicated in Fig. 6.

Table 4. Domain formalization of the solar system (S) and of Rutherford's atom model (T) as used by HDTP.

Sorts:
<i>real, object, time</i>
Entities:
<i>sun, planet, nucleus, electron : object</i>
Shared functions of both theories:
<i>mass : object \rightarrow real \times {kg} dist : object \times object \times time \rightarrow real \times {m}</i>
Functions of the solar system theory:
<i>force : object \times object \times time \rightarrow real \times {N} gravity : object \times object \times time \rightarrow real \times {N}</i>
<i>centrifugal : object \times object \times time \rightarrow real \times {m}</i>
Predicates of the solar system theory:
<i>revolves_around : object \times object</i>
Facts of the solar system theory:
<i>(α_1) mass(sun) > mass(planet) (α_2) mass(planet) > 0 (α_3) $\forall t : \text{time} : \text{gravity}(\text{planet}, \text{sun}, t) > 0$</i>
<i>(α_4) $\forall t : \text{time} : \text{dist}(\text{planet}, \text{sun}, t) > 0$</i>
Laws of the solar system theory:
<i>(α_5) $\forall t : \text{time}, o_1 : \text{object}, o_2 : \text{object} : \text{dist}(o_1, o_2, t) > 0 \wedge \text{gravity}(o_1, o_2, t) > 0 \rightarrow \text{centrifugal}(o_1, o_2, t) = -\text{gravity}(o_1, o_2, t)$</i>
<i>(α_6) $\forall t : \text{time}, o_1 : \text{object}, o_2 : \text{object} : 0 < \text{mass}(o_1) < \text{mass}(o_2) \wedge \text{dist}(o_1, o_2, t) > 0 \wedge \text{centrifugal}(o_1, o_2, t) < 0$</i>
<i>$\rightarrow \text{revolves_around}(o_1, o_2)$</i>
Functions of the atom model theory:
<i>coulomb : object \times object \times time \rightarrow real \times {N}</i>
Facts of the atom model theory:
<i>(β_1) mass(nucleus) > mass(electron) (β_2) mass(electron) > 0 (β_3) $\forall t : \text{time} : \text{coulomb}(\text{electron}, \text{nucleus}, t) > 0$</i>
<i>(β_4) $\forall t : \text{time} : \text{dist}(\text{electron}, \text{nucleus}, t) > 0$</i>

Table 4 gives the initial domain formalizations, containing a governing theory of the solar system S and an account of Rutherford's alleged knowledge T about some aspects relating to the atom's structure. When provided with these inputs, HDTP is used to compute a shared generalization G between S and T , together with the corresponding sets of substitutions ϕ_S and ϕ_T for re-obtaining the covered domain parts $S_c \subseteq S$ and $T_c \subseteq T$. Given ϕ_S , the set of anti-unifications inversely corresponding to the respective substitutions is used for obtaining the generalized source theory S' (given in Table 5), not only containing elements of the generalization corresponding to axioms from S_c but also generalizing axioms from $S \setminus S_c$. Also notice that for the given formalizations of S and T it holds that $T = T_c$, i.e., the target theory is entirely covered by the common generalization G .

The generalized source theory S' is now used to compute an asymmetric amalgam with (the unchanged) domain theory T , pairwise unifying axioms from S' with T . Due to T being completely covered by the common generalization, all elements from T can be directly unified with elements from S' , leaving out only the definition of the *centrifugal* function, the *revolves_around* predicate, and the laws γ_5 and γ_6 on the side of S' . These axioms are then added as additional elements to the unified set of axioms, resulting in the proto-blend theory T' . As γ_5 and γ_6 contain non-instantiated variables obtained by generalizing S into S' , we now apply ϕ_T to T' , resulting in the final blended theory T_B , giving an account of the atom model with the electrons revolving around the nucleus in stable orbits due to the equality between coulomb and centrifugal force. A consistency check of T then confirms that the resulting blend is consistent and, thus, the procedure terminates.

Table 5. Generalized source theory S' based on the common generalization between the solar system and the Rutherford atom (axioms not obtained from the covered subset S_c are highlighted by *).

Sorts:
<i>real, object, time</i>
Entities:
<i>X, Y : object</i>
Functions:
<i>mass : object \rightarrow real \times {kg} dist : object \times object \times time \rightarrow real \times {m}</i>
<i>F : object \times object \times time \rightarrow real \times {N} centrifugal : object \times object \times time \rightarrow real \times {m}</i>
Predicates:
<i>revolves_around : object \times object</i>
Facts:
<i>(γ_1) mass(X) > mass(Y) (γ_2) mass(Y) > 0 (γ_3) $\forall t : time : F(X, Y, t) > 0$</i>
<i>(γ_4) $\forall t : time : dist(X, Y, t) > 0$</i>
Laws:
<i>(γ_5^*) $\forall t : time, o_1 : object, o_2 : object : dist(o_1, o_2, t) > 0 \wedge F(o_1, o_2, t) > 0 \rightarrow centrifugal(o_1, o_2, t) = -F(o_1, o_2, t)$</i>
<i>(γ_6^*) $\forall t : time, o_1 : object, o_2 : object : 0 < mass(o_1) < mass(o_2) \wedge dist(o_1, o_2, t) > 0 \wedge centrifugal(o_1, o_2, t) < 0$</i>
<i>$\rightarrow revolves_around(o_1, o_2)$</i>

Clearly, this example stays very close to the classical analogy case. Still, we maintain that the concept blending perspective also offers a valid view on what is going on as the combination of the initial observations about a few aspects of the inner workings of the atom and the governing theory of the solar system adds previously unconsidered and at first not directly connected aspects to the former: the axiom about stable orbits for atoms σ_6 can only be meaningfully integrated due to the presence of the (also new) axiom σ_5 , establishing the correspondence relationship between centrifugal and coulomb force. Thus, we can see a qualitative change and a significant step of contentual evolution between T and T_B

3.4 Example 2: Pegasus as Mythological Concept Blend

One of the best known concept blends is Pegasus, the winged divine stallion and son of Poseidon and the Gorgon Medusa from classical Greek mythology. From a concept blending perspective, Pegasus constitutes a blend between a stereotypical horse and a stereotypical bird, maintaining all the horse characteristics but adding bird-like features such as, for instance, the wings and the ability to fly. In what follows, we will re-construct the blending process underlying Pegasus' concept formation as second application example for the analogy-based perspective on blending in COINVENT.

We start with the stereotypical characterizations of a horse and a bird in a many-sorted first-order logic representation (as used by HDTP) from Table 6. Given these characterizations, HDTP can be used for finding a common generalization of both, subsequently again using the anti-unifications corresponding to ϕ_S for generalizing the source theory S into S' as given in Table 7. Computing the asymmetric amalgam of S' with the (fixed) target theory T , we obtain the proto-blend T' from Table 8. As T' still features axioms containing non-instantiated variables, ϕ_T is applied to the theory resulting in the (with respect to ϕ_T) fully instantiated blend theory T_B from Table 9.

In a final step, a consistency check of the blended theory T_B is performed. Taking into account world knowledge about mammals causes a clash with axiom δ_7 as mammals generally do not lay

Table 6. Example formalizations of stereotypical characterizations for a bird S and a horse T .

Sorts:
clade, entity, bodypart, ability

Entities:
mammal, avialae : clade horse, bird : entity torso, legs, wings : bodypart walk, fly, lay_eggs : ability

Predicates:
is_of_clade : entity \times clade, has_bodypart : entity \times bodypart, has_ability : entity \times ability

Laws of the bird characterization:
(α_1) is_of_clade(bird, avialae) (α_2) has_bodypart(bird, legs) (α_3) has_bodypart(bird, torso)
(α_4) has_bodypart(bird, wings) (α_5) has_ability(bird, walk) (α_6) has_ability(bird, fly)
(α_7) has_ability(bird, lay_eggs)

Laws of the horse characterization:
(β_1) is_of_clade(horse, mammal) (β_2) has_bodypart(horse, legs) (β_3) has_bodypart(horse, torso)
(β_4) has_ability(horse, walk)

 Table 7. Abbreviated representation of the generalized source theory S' based on the stereotypical characterizations for a horse and a bird (axioms not obtained from the covered subset S_c are highlighted by *).

Entities:
C : clade, E : entity

Laws:
(γ_1) is_of_clade(E, C) (γ_2) has_bodypart(E, legs) (γ_3) has_bodypart(E, torso) (γ_4) has_ability(E, walk)
(γ_5^) has_bodypart(E, wings) (γ_6^*) has_ability(E, fly) (γ_7^*) has_ability(E, lay_eggs)*

 Table 8. Abbreviated representation of the proto-blend T' obtained from computing the asymmetric amalgam between S' and T (axioms from S' which had not been obtained from S_c are highlighted by *).

Entities:
E : entity

Laws:
(δ_1) is_of_clade(horse, mammal) (δ_2) has_bodypart(horse, legs) (δ_3) has_bodypart(horse, torso)
(δ_4) has_ability(horse, walk) (δ_5^) has_bodypart(E, wings) (δ_6^*) has_ability(E, fly) (δ_7^*) has_ability(E, lay_eggs)*

 Table 9. Abbreviated representation of $T_B = \phi_T(T')$ (axioms which have been instantiated from axioms that had originally not been obtained from S_c are highlighted by *).

Laws:
(δ_1) is_of_clade(horse, mammal) (δ_2) has_bodypart(horse, legs) (δ_3) has_bodypart(horse, torso)
(δ_4) has_ability(horse, walk) (δ_5^) has_bodypart(horse, wings) (δ_6^*) has_ability(horse, fly)*
(δ_7^) has_ability(horse, lay_eggs)*

eggs. Thus, returning to the start of the procedure, we can re-initiate the algorithm, for example, with $S_{clash} = S \setminus \{\alpha_7\}$, and finally obtain the (with respect to ϕ_T fully instantiated and consistent) version of T_B given in Table 10 as output.

Table 10. Abbreviated representation of the final blended theory T_B after inconsistency check and repair (i.e., based on $S_{clash} = S \setminus \{\alpha_7\}$).

Laws:

(δ_1) <i>is_of_clade</i> (horse, mammal)	(δ_2) <i>has_bodypart</i> (horse, legs)	(δ_3) <i>has_bodypart</i> (horse, torso)
(δ_4) <i>has_ability</i> (horse, walk)	(δ_5) <i>has_bodypart</i> (horse, wings)	(δ_6) <i>has_ability</i> (horse, fly)

4. Conclusions

In the previous sections we presented an analogy-inspired perspective on and a working algorithm for the blending of concept theories taken from ongoing research within the COINVENT project. The main contributions are the combination of generalization-based analogy and (a restricted version of) the HDTP system with a well-founded formal model and mechanism for knowledge transfer and concept combination in form of the amalgam framework: Building upon HDTP’s approach to generalization and domain matching asymmetric amalgams allow to soundly compute the concept blend of two input theories in a controlled fashion.

The next step on the formal side in the development of our concept blending framework (other than the aspects already mentioned in Sect. 3.2) will be to remove the restriction on the substitutions used in HDTP, allowing for applications of all four types of substitutions admissible in restricted higher-order anti-unification. If this constraint is lifted, a replacement for the semantic consequence relationship \models as basis for the subsumption ordering will have to be found. Here, we hope that providing a semantics to the syntax-based operations in HDTP and restricted higher-order anti-unification via derived signature morphisms (Krumnack, Mossakowski, & Maibaum, 2014) will allow us to subsequently construct a suitable substitute.

Clearly, ours is by far not the only attempt at the computational modeling of concept blending, nor is it the only point of view represented within COINVENT: For instance Martinez et al. (2014) recently presented an approach for algorithmic blending of mathematical theories also building upon Goguen (2006)’s ideas, Kutz et al. (2015) give an account and a system model of computer-based blending of ontologies, and Li et al. (2012) report on case studies of systems taking into account goals and contexts in the blending-based production of creative artifacts. Still, in combining the generality of modeling introduced by HDTP’s use of first-order logic languages with the formal soundness and solid theoretical foundations of the underlying generalization model and the amalgam framework, the envisioned COINVENT system should constitute a valuable contribution to the ongoing discourse and a big step towards a computationally feasible model of concept blending.

Acknowledgements

The authors acknowledge the financial support of the Future and Emerging Technologies programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 611553 (COINVENT).

References

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7, 39–59.
- Besold, T. R. (2014). Sensorimotor Analogies in Learning Abstract Skills and Knowledge: Modeling Analogy-Supported Education in Mathematics and Physics. *Proc. of the AAAI Fall 2014 Symposium on Modeling Changing Perspectives: Reconceptualizing Sensorimotor Experiences*.
- Boden, M. A. (2003). *The creative mind: Myths and mechanisms*. Routledge.
- Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41, 1–63.
- Fauconnier, G., & Turner, M. (1998). Conceptual integration networks. *Cognitive Science*, 22, 133–187.
- Gentner, D., & Smith, L. A. (2013). Analogical learning and reasoning. In D. Reisberg (Ed.), *The Oxford handbook of cognitive psychology*, 668–681. Oxford University Press.
- Goguen, J. (2006). Mathematical models of cognitive space and time. *Reasoning and Cognition; Proc. of the Interdisciplinary Conference on Reasoning and Cognition* (pp. 125–128).
- Goguen, J. A., & Harrell, D. F. (2010). Style: A computational and conceptual blending-based approach. In S. Argamon, K. Burns, & S. Dubnov (Eds.), *The structure of style*, 291–316. Springer.
- Guhe, M., Pease, A., Smaill, A., Schmidt, M., Gust, H., Kühnberger, K.-U., & Krumnack, U. (2010). Mathematical reasoning with higher-order anti-unification. *Proc. of the 32nd Annual Conference of the Cognitive Science Society* (pp. 1992–1997). Cognitive Science Society.
- Hofstadter, D., & Mitchell, M. (1994). The copycat project: a model of mental fluidity and analogy-making. *Advances in Connectionist and Neural Computation Theory* (pp. 31–112). Ablex.
- Krumnack, U., Mossakowski, T., & Maibaum, T. (2014). What is a derived signature morphism? *Preliminary Proc. of WADT 2014: 22nd International Workshop on Algebraic Development Techniques* (pp. 50–51). Simion Stoilow Institute of Mathematics of the Romanian Academy.
- Kutz, O., Bateman, J., Neuhaus, F., Mossakowski, T., & Bhatt, M. (2015). E pluribus unum. In T. R. Besold, M. Schorlemmer, & A. Smaill (Eds.), *Computational creativity research: Towards creative machines*, Vol. 7 of *Atlantis Thinking Machines*, 167–196. Atlantis Press.
- Li, B., Zook, A., Davis, N., & Riedl, M. (2012). Goal-driven conceptual blending: A computational approach for creativity. *Proc. of the Third International Conference on Computational Creativity* (pp. 9–16).
- Martinez, M., Krumnack, U., Smaill, A., Besold, T. R., Abdel-Fattah, A. M., Schmidt, M., Gust, H., Kühnberger, K.-U., Guhe, M., & Pease, A. (2014). Algorithmic aspects of theory blending. In G. Aranda-Corral, J. Calmet, & F. Martín-Mateos (Eds.), *Artificial intelligence and symbolic computation*, Vol. 8884 of *Lecture Notes in Computer Science*, 180–192. Springer.
- Ontañón, S., & Plaza, E. (2010). Amalgams: A formal approach for combining multiple case solutions. In I. Bichindaritz & S. Montani (Eds.), *Case-Based Reasoning: Research and Development*, Vol. 6176 of *Lecture Notes in Computer Science*, 257–271. Springer.

- Ontañón, S., & Plaza, E. (2012). On knowledge transfer in case-based inference. In B. D. Agudo & I. Watson (Eds.), *Case-based reasoning research and development*, Vol. 7466 of *Lecture Notes in Computer Science*, 312–326. Springer.
- Pereira, F. C. (2007). *Creativity and AI: A conceptual blending approach*. Mouton de Gruyter.
- Plotkin, G. D. (1970). A note on inductive generalization. *Machine Intelligence*, 5, 153–163.
- Schmidt, M., Krumnack, U., Gust, H., & Kühnberger, K.-U. (2014). Heuristic-driven theory projection: An overview. In H. Prade & G. Richard (Eds.), *Computational approaches to analogical reasoning: Current trends*, 163–194. Springer.
- Schwering, A., Gust, H., & Kühnberger, K.-U. (2009). Solving Geometric Analogies with the Analogy Model HDTP. *Proc. of the 31st Annual Meeting of the Cognitive Science Society* (pp. 1780–1785). Cognitive Science Society.
- Schwering, A., Krumnack, U., Kühnberger, K.-U., & Gust, H. (2009). Syntactic principles of Heuristic-Driven Theory Projection. *Journal of Cognitive Systems Research*, 10, 251–269.
- Thagard, P., & Stewart, T. C. (2011). The aha! experience: Creativity through emergent binding in neural networks. *Cognitive Science*, 35, 1–33.
- Veale, T., & O’Donoghue, D. (2000). Computation and blending. *Cognitive Linguistics*, 11, 253–281.
- Winston, P. H. (1980). Learning and reasoning by analogy. *Commun. ACM*, 23, 689–703.