



City Research Online

City St George's, University of London

Citation: Besold, T. R., Kuhnberger, K-U. & Plaza, E. (2017). Towards a computational- and algorithmic-level account of concept blending using analogies and amalgams. *Connection Science*, 29(4), pp. 387-413. doi: 10.1080/09540091.2017.1326463

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/18664/>

Link to published version: <https://doi.org/10.1080/09540091.2017.1326463>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

To appear in *Connection Science*
Vol. 00, No. 00, Month 20XX, 1–26

ORIGINAL ARTICLE

Towards a Computational- and Algorithmic-Level Account of Concept Blending Using Analogies and Amalgams

Tarek R. Besold^{a*} and Kai-Uwe Kühnberger^b and Enric Plaza^c

^a*Digital Media Lab, Center for Computing and Communication Technologies (TZI), University of Bremen, Germany;* ^b*Institute of Cognitive Science, University of Osnabrück, Germany;*

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

(August 2, 2016)

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 article we summarize our work towards the development of a computational-level and algorithmic-level account of concept blending, combining approaches from computational analogy-making and case-based reasoning (CBR). We present the theoretical background, as well as an algorithmic proposal integrating higher-order anti-unification matching and generalization from analogy with amalgams from CBR. The feasibility of the approach is then exemplified in two case studies.

Keywords: Concept Blending; Cognitive Artificial Intelligence; Computational Creativity; Analogy; Amalgams

1. Introduction: Computational Creativity and Concept Blending

M. A. 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 modelling 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 characterisation be given, nor are the details of a possible computational-level theory describing the process(es) at work well understood.¹ Still, in recent years a proposal by Fauconnier and Turner (1998) called

*Corresponding author. Email: tbesold@uni-bremen.de

¹In the title and throughout the article, the terms “algorithmic characterisation”, “computational-level theory”,

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. Nevertheless, Fauconnier and Turner (also in their later works) remain mostly silent concerning details needed for a proper computational modelling of concept blending as cognitive capacity — neither do they provide a fully worked out and formalised 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 spaces 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 modelling have used the provided conceptual descriptions as starting point for suggesting possible refinements and implementations: J. A. Goguen and 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 hypothesised sub-parts of concept blending, Thagard and 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 and O'Donoghue (2000) present their computational model of conceptual integration and propose several extensions to the at the time actual view on concept blending.

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 had been undertaken in the EU-FP7 “Concept Invention Theory” (COINVENT) project.² One of the main goals of the COINVENT research program was the development of a computational-level and algorithmic-level account of concept blending based on insights from psychology, AI, and cognitive modelling, 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 resulting COINVENT core model for concept blending and show how the respective mechanisms and systems interact.

2. Cognitive Task and Theoretical Commitments

The cognitive task targeted by COINVENT was the blending of conceptual theories, i.e., of logic-based representations of real-world concepts in finite axiomatisations. Contrary to accounts of mathematical theory blending as, e.g., presented by Martinez et al. (2014), a conceptual theory is not necessarily a (modulo logical equivalence) unique and unambiguous description of a concept. Mathematical theories can offer both properties due to the axiomatic nature of mathematical concepts, making them conceivable as being made up exclusively by the set of defining axioms (or the deductive closure thereof). In contrast, conceptual theories as descriptive results of a formalisation process in most cases

etc. refer to the corresponding conceptualisations in Marr (1982)'s Tri-Level Hypothesis.

²Cf. <http://www.coinvent-project.eu>.

represent a certain partial perspective (among several possible ones) on the described concept, instantiating what Bou et al. (2015) call a conceptual space:

“Conceptual spaces are partial and temporary representational structures which are constructed on the fly when talking about a particular situation, which are informed by the knowledge structures associated with a domain. These are influenced by Boden’s idea of a concept space which is mapped, explored and transformed by transcending mapped boundaries (M. Boden, 1977) (...).” (Bou et al., 2015, p. 56)

Still, with respect to the computational-level account of the cognitive task addressed by our theory and system, strong parallels exist to the work on mathematical theory blending. The system is presented with two input theories formalised as finite axiomatisations in possibly different logic-based languages (in the case of the algorithmic-level system description in Section 4.3: many-sorted first-order logics), and produces as output another theory describing the blend between the formalised input concepts again as finite axiomatisation in a logic-based language.

In terms of the corresponding account of creativity, we thereby aim to automatise an important mechanism as part of combinatorial creativity in Boden’s sense. Approaching this (family of) task(s) via theory blending allows to accommodate for several theoretical characteristics we perceive to be central to blending on the level of real-world concepts:

- *Concept blending*: Blending happens on the knowledge level. While people as cognitive agents in most cases might not be aware of it, the blending process crucially relies on knowledge about the input concepts available to the cogniser. The blending process is then guided by similarities between the input concepts, as characteristics of concepts are not arbitrarily combined during the blending process, but the blending process is guided by shared properties/elements of the input concepts. These similarities also define the basic structure of the resulting blend(s).
- *Similarities, analogy, and amalgams*: Similarities between input concepts are accessible via meaningful generalisation between concepts. On the level of conceptual theories, this corresponds to the anti-unification of theories (see Section 3.1 for details). As such, generalisation-based analogy-making is a suitable approach for identifying and subsequently (via analogical transfer) carrying over these similarities into the basic structure of the blend. The combination of further properties from both input theories can then be conceived of as generalisation-based amalgamation, maintaining the basic structure introduced by the analogy process.
- *Constraints on the blending process*: On the system side, further external constraints imposed by the environment and/or task, and internal properties of the cognitive agent (such as, e.g., expertise) can be taken into account through heuristics and knowledge-sensitive methods during the computation of the blend(s) and the subsequent selection of the final output theory.

Given the central role generalisation, analogy-making, and amalgamation play in our understanding and computational-level theorising of the blending of conceptual theories, we proceed with fairly detailed introductions to the accounts of computational analogy-making and the computation of amalgams used in our theory and system (which are then described in detail in Section 4).

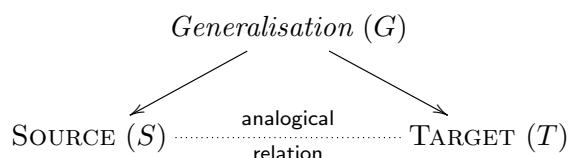


Figure 1. A schematic overview of HDTP's generalisation-based approach to analogy.

3. Computational Models of Analogy and Amalgams

As analogy seems to play a crucial role in human cognition (see, for instance, the overview provided by Gentner and 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 Falkenhainer, Forbus, and Gentner (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, Krumnack, Gust, and Kühnberger (2014)), a generalisation-based symbolic analogy engine discussed in detail in Sect. 3.1.

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. As described by Aamodt and Plaza (1994), 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. 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 and Plaza (2012) outlining a recent attempt at also gaining insights on this part of the CBR cycle by suggesting the framework of amalgams (originally introduced by Ontañón and Plaza (2010)) as a formal model. Sect. 3.2 gives an overview of amalgams as used in the COINVENT model.

3.1. *Representing and Computing Generalisation-Based Analogies 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 axiomatisations, i.e., given by a finite set of formulae. HDTP follows a generalisation-based approach to analogy-making: given both domains, a common generalisation encompassing structurally shared elements common to both input domains is computed (*mapping phase*) and this generalisation 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.

Sorts:	<i>clade, entity, bodypart, ability</i>
Entities:	<i>mammal : clade horse : entity torso, legs : bodypart walk : ability</i>
Predicates:	<i>is_of_clade : entity × clade has_bodypart : entity × bodypart has_ability : entity × ability</i>
Laws of the horse characterisation:	<i>is_of_clade(horse, mammal) has_bodypart(horse, legs) has_bodypart(horse, torso) has_ability(horse, walk)</i>

Table 1. Example formalisation of a stereotypical characterisation of a horse.

3.1.1. Representing Domain Theories and Generalisation Steps Between Domains in HDTP

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 $\text{Term}(\Sigma, \mathcal{V})$ and the function $\text{sort} : \text{Term}(\Sigma, \mathcal{V}) \rightarrow \text{Sort}$ are defined inductively as follows:

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

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

As an example for a domain representation using HDTP’s language, Table 1 reproduces a possible formalisation of the concept of “horse” using some of a horse’s key characteristics (this formalisation reoccurs below as part of a bigger case study demonstrating the concept blending capacities of the framework combining analogy and amalgams in Section 4.4):

Given two input domains, HDTP uses anti-unification (firstly studied in a first-order setting by Plotkin (1970)) to compute a generalisation of both domains. In this process, terms are generalised 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. These “replacements” can be formalised as substitutions:

DEFINITION 3 Given term algebra $\text{Term}(\Sigma, \mathcal{V})$. A term substitution is a partial function $\sigma : \mathcal{V} \rightarrow \text{Term}(\Sigma, \mathcal{V})$ 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, generalisations can formally be characterised, with the *least general generalisation* playing a special role as most specific anti-unifier (i.e., as minimal with respect to the instantiation order):

DEFINITION 4 Let f, g be terms from a term algebra $Term(\Sigma, \mathcal{V})$. A generalisation 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 generalisation $\langle G, \sigma, \tau \rangle$ is called the least general generalisation (LGG) if for any generalisation $\langle G', \sigma', \tau' \rangle$ 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.

3.1.2. Computing Least General Generalisations Using Restricted Higher-Order Anti-Unification as Basis for Analogies in HDTP

Against this background, Schwering, Krumnack, Kühnberger, and Gust (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 (also see Fig. 2 for concrete examples of the defined substitution operations).

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' .

Clearly, restricted substitutions are strictly more general than mere (first-order) term

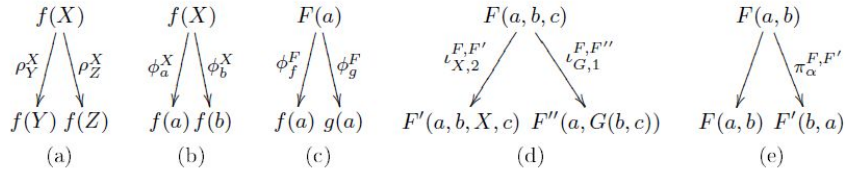


Figure 2. A reproduction of the examples originally given by Schwering et al. (2009) for the different types of higher-order anti-unifications applied in HDTP: A renaming (a), two different forms of fixation (b and c), an argument insertion (d), and a permutation (e).

Sorts:
clade, entity, bodypart, ability

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

Predicates:
*is_of_clade : entity × clade has_bodypart : entity × bodypart
 has_ability : entity × ability*

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

Table 2. Example formalisation of a stereotypical characterisation of a dog.

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 generalisations according to a complexity order on the complexity of generalisation (based on a complexity measure for substitutions), and finally chooses the least complex generalisations 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 generalisation 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 formulae 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 generalisation 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” formalisation from Table 1 and the stereotypical characterisation of a dog given in Table 2, and generalise them into the shared generalisation in Table 3.

When processing sets of formulae, a heuristic is applied for iteratively selecting pairs of formulae to be generalised: 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 maximise the coverage of generalisations:

Sorts:
<i>clade, entity, bodypart, ability</i>
Entities:
<i>mammal : clade E : entity B : bodypart A : ability</i>
Predicates:
<i>is_of.clade : entity × clade has_bodypart : entity × bodypart</i>
<i>has_ability : entity × ability</i>
Laws of the shared characterisation:
<i>is_of.clade(E, mammal) has_bodypart(E, B) has_ability(E, A)</i>

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

DEFINITION 7 *Given a generalisation $\langle G, \sigma, \tau \rangle$ 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 generalisation $\langle G, \sigma, \tau \rangle$ 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 generalisations).

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

Once obtained, the generalised 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.

3.2. Representing and Guiding the Combination of Conceptual Theories Using Amalgams

Ontañón and Plaza (2010) developed the notion of amalgams in the context of Case-based Reasoning (CBR), where new problems are tackled 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 address the new problem. The notion of an “amalgam” of two cases (two descriptions of problems and their solutions) is a proposal to formalise the ways in which cases can be combined to produce a new, coherent case.

Formally, amalgams can be defined in any representation language \mathcal{L} for which a subsumption relation \sqsubseteq between the formulae (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 .³ 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 specialisation of two given descriptions, and their anti-unification, $(I_1 \sqcap I_2)$, defined as the least general generalisation 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 $I_1 \sqcup I_2 = \top$, i.e.,

³In machine learning terms, $A \sqsubseteq B$ means that A is more general than B , while in description logics it has the opposite meaning, since it is seen as ‘set inclusion’ of their interpretations.

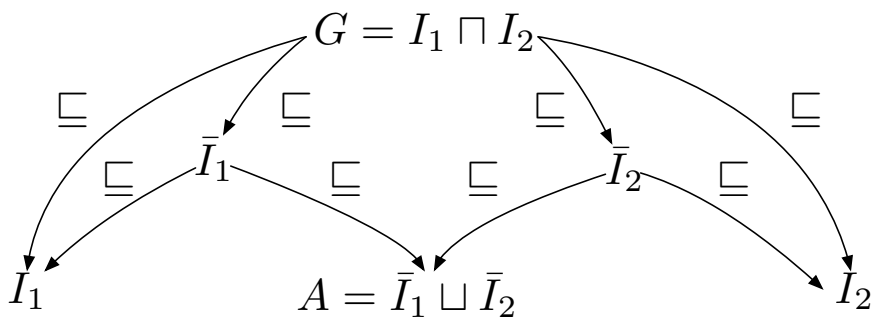


Figure 3. A diagram of an amalgam A from inputs I_1 and I_2 where $A = \bar{I}_1 \sqcup \bar{I}_2$.

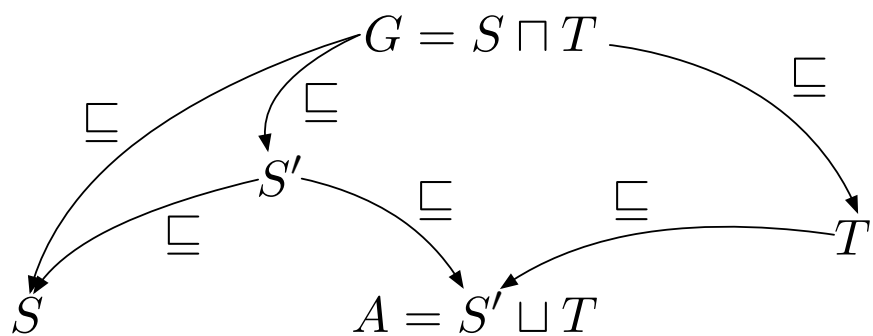


Figure 4. A diagram that transfers content from source S to a target T via an asymmetric amalgam A .

they have no common specialisation 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.

Amalgams can be conceived of as a generalisation of the notion of unification: as ‘partial unification’ (see the description given by Ontañón and Plaza (2010) for details). Unification means that what is true for I_1 or I_2 is also 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 specialisation 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 specialisation except \top . An amalgam of two descriptions is a new description that contains parts from each of the two original 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 article 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 generalisations \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 \bar{I}_1 and \bar{I}_2 have no common specialisation then trivially $A = \top$, since their only unifier is ‘none’. For our purpose we will only be 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 generalisations \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 (i.e. all descriptions that are consistent and can be defined as the unification of two generalisations of the inputs).

Usually we are interested only in 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 generalising 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 generalised, the target is not. As we shall see, asymmetric amalgams share important properties with analogical inference: while the source can be relaxed and thus lose information, the target is fixed, so all information belonging to the target will be present in the final (asymmetric) amalgam.

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 generalisation of the source $S' \sqsubseteq S$.*

As shown in Fig. 4, the content of target T is transferred completely into the asymmetric amalgam, while the source S is generalised. The result is a form of partial unification that preserves all the information in T while relaxing S by generalisation and then unifying one of those generalisations S' with T . 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 this reason, asymmetric amalgams 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 and Plaza (2012).

4. The COINVENT Account of Analogy-Based Concept Blending

The previous section gave an introduction to computational analogy-making using HDTP and to the combination of conceptual theories within the framework of amalgamation. Both theories and the corresponding mechanisms underlie the COINVENT model for concept blending and its proof-of-concept implementation described in this section. Naturally, ours is not the only attempt at modelling and automatizing concept blending capacities in a computational system. In addition to those already mentioned in Section 1, a short overview of other efforts aiming to achieve similar goals is given in Section 5.

The following Section 4.1 provides an introduction to the COINVENT model, starting with a computational-level description before pushing towards the level of detail needed for the algorithmic-level implementation described in Section 4.3. Section 4.2 gives an example for the computational-level dynamics, by this grounding the previous mostly abstract descriptions, while Section 4.4 and 4.5 detail two prototypical application case studies for the proposed implementation of the COINVENT model.

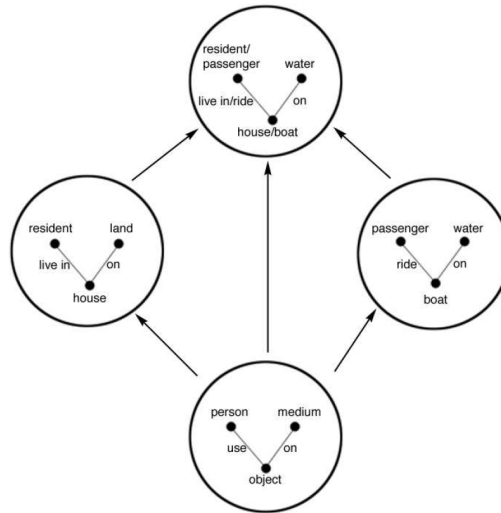


Figure 5. Schematic overview of the houseboat blend as conceptualised by J. Goguen (2006): The conceptual theories for house and boat are generalized to a theory describing some object used by a person resting on some medium, and then combined to a houseboat theory featuring an object which is at the same time house and boat, resting on water, with residents living in it (who are at the same time passengers riding on it).

4.1. The COINVENT Model of Concept Blending

One of the early formal accounts on concept blending—which is especially influential to the approach in COINVENT—is the classical work by J. Goguen (2006) using notions from algebraic specification and category theory. This version of concept blending can be described by the diagram in Fig. 6, where each node stands for a representation an agent has of some concept or conceptual domain. As stated in Section 2, we consider these representations to be conceptual theories, corresponding to 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 theories (and associated spaces). The idea is that, given two conceptual theories I_1 and I_2 as input, we look for a generalisation G and then construct a blend space B in such a way as to preserve as many as possible of the structural alignments between I_1 and I_2 established by the generalisation. As an example, consider J. Goguen (2006)’s houseboat blend depicted in Figure 5.

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 . Again using an example by J. Goguen (2006), this would for example become relevant in the case of blending a house and a boat into a boat used as a land-based shelter: Given the input theories in Figure 5, in that case the blend theory would have to omit axioms putting the house/boat on water, and stating that a passenger rides aboard the house/boat. In any case, as the blend respects (to the largest possible extent) the relationship between I_1 and I_2 , the diagram will commute. Clearly, this approach is structurally similar to the mechanism underlying generalisation-based analogy-making in HDTP (with the latter only missing the blending capabilities; also compare Fig. 6 and Fig. 1).

Concept invention by concept blending can then be phrased as the following task: given two axiomatisations of two domain theories I_1 and I_2 , we need first, to compute

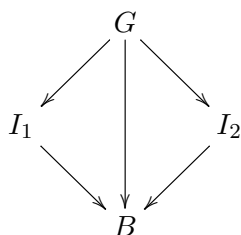


Figure 6. A conceptual overview of J. Goguen (2006)’s account of concept blending.

a generalised 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 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 .⁴

Provided with S and T , according to the just given task description, the following steps have to be accounted for:

- (1) Find a joint generalisation G between the input domains S and T , accounting for the shared structure between both.
- (2) Building on G , fully generalise S .
- (3) Combine the generalised version of S in an asymmetric amalgam with T to obtain the blend T_B .
- (4) Assure consistency of T_B and output the blend.

In the first step, the reasoning process is triggered by the computation of the generalisation G (generic space). When using an analogy engine for this purpose, for concept invention we will only need the mapping mechanism and replace the transfer phase (which is characteristic for analogy-making) by a new blending algorithm. In the case of HDTP, the mapping is achieved via the usual generalisation process between S and T , in which pairs of formulae from the source and target theories are anti-unified resulting in a generalised theory that reflects common aspects of both spaces. The generalised theory can be projected into the original ones 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 Section 3.1). 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 generalisation 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: in the second step, the set of substitutions

⁴In the case where additional properties are provided by both domains the same general principles as described below apply. It just becomes necessary to also treat the target domain T similar to the current source S , expanding the conceptual overview in Fig. 7 with a second “generalisation triangle” to the right of the “blending diamond”, computing a generalisation of T and using the latter for the blending process (for which only minor and quite straightforward changes become necessary, assuring that all terms in the resulting blend are grounded and no variables introduced during the generalisation steps remain uninstantiated).

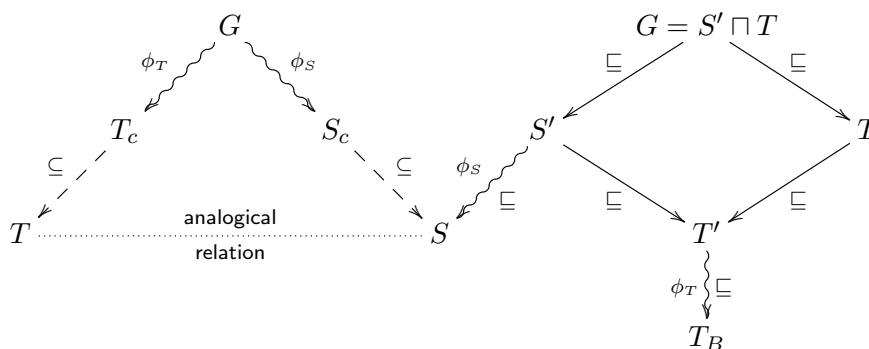


Figure 7. A conceptual overview of the COINVENT model of concept blending as described in Section 4.1: The shared generalisation G from S and T is computed with $\phi_S(G) = S_c$. The relation ϕ_S is subsequently re-used in the generalisation of S into S' , which is then combined in an asymmetric amalgam 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.)

can be inverted and applied to generalise the original source theory S into a more general version S' (forming a superset of the shared generalisation G , also including previously uncovered knowledge from the source) which then can be combined in the third step into an asymmetric amalgam with the target theory T , forming the (possibly underspecified) proto-blend T' of both. Concluding this step and also the blending process itself, T' is then completed into the blended theory and output of the process T_B by applying corresponding specialisation steps stored from the generalisation process between S and T (see also Fig. 7). In the final step, the resulting blend theory T_B then is checked for consistency in the logical sense and regarding potentially available world knowledge, since inconsistencies of either type could have been introduced when transferring axioms into the blend. If inconsistencies are detected, repair mechanisms are triggered until a consistent blend theory is returned as final output of the blending process.

At this point a remark concerning the selection of good blends among the many possible ones is in place. As should also become obvious from the conceptual overview, there are several stages in the process where implementation details will significantly influence the precise outcome of the blending process (e.g., a change in HDTP's heuristics could significantly change the outcome of the selection of T_c and S_c , and consequently all subsequent steps). Assessing the quality of a blend is generally considered a hard task since the evaluation strongly depends, among others, on the context and the purpose of the blending process. The selection of particularly good blends—or, more generally, of specific types of blends over others—has therefore not been in the focus of this work. Still, it has been treated in other places both within the COINVENT project, as well as by other researchers. A first set of informal and heuristic style optimality principles for concept blends had been postulated by Fauconnier and Turner (1998). Unfortunately, while being helpful especially in the evaluation of linguistic blends, they lack the precision as to directly be algorithmically realizable. Pereira and Cardoso (2003) attempted to fill this gap by proposing one possible implementation. As part of COINVENT, Confalonieri, Corneli, Pease, Plaza, and Schorlemmer (2015) suggested to use computational argumentation for evaluating concept blends (as well as other forms of combinatorial creativity). In their approach, concept blends are evaluated in an open-ended and dynamic discussion, allowing for the improvement of blends and the explicit representation of the reasons behind an evaluation. Relatedly, Schorlemmer, Confalonieri, and Plaza (2016)

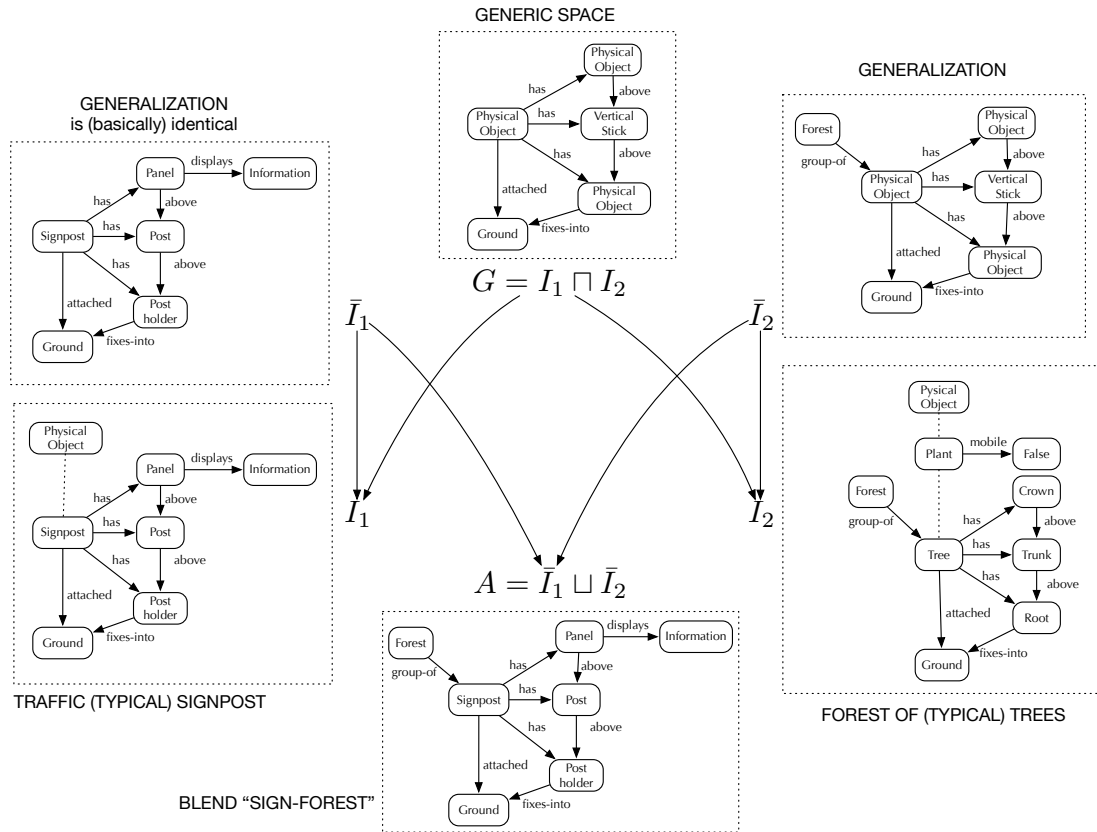


Figure 8. Blending schema for “Sign Forest” when inputs are typical concepts for “Sign” (traffic signpost) and “Forest” (forest of typical trees); the arrows indicate subsumption (\sqsubseteq) as in Figure 3.

discussed how newly invented concepts can be evaluated with respect to a background ontology of conceptual knowledge, checking which can be added to the system of familiar concepts, and how the previously given conceptualisation might be altered.

4.2. Example: Following Trees and Signposts into the Sign Forest

In this first example, originally introduced by Besold and Plaza (2015), we want to illustrate the overall approach to generalisation-based blending using analogy and amalgamation. In this section we therefore leave aside the representational and algorithmic characteristics imposed by the use of HDTP as specific generalisation mechanism and analogy-engine (which will be elaborated upon in detail in the actual account of the system given in Section 4.3, and also in the later case studies in Section 4.4 and Section 4.5), but focus exclusively on the high-level dynamics. To this end, an ontology-driven approach to generalisation is taken instead, allowing for intuitive and straightforward “semantic” generalisations and analogies within a common concept hierarchy.

We reconstruct the sign forest blend discussed in Kutz, Mossakowski, Hois, Bhatt, and Bateman (2012), providing an interpretation of the concept from a metaphor-centred perspective, and show how the general COINVENT model can serve for modelling the blending process. In what follows we consider sign forest equivalent to the (interpretation of the concept detailed by the) expression “a forest of signs”, that shows more clearly its metaphorical nature.

The concept of tree (and, thus, our corresponding conceptual theory) is typically conceived as a plant having roots, a trunk and a crown (even if there may be plants categorised as trees that do not have a trunk, this is ignored as it does not belong to the bundle of properties that are typical); this view is depicted as I_2 in the bottom right of Fig. 8, where other properties are included, like plants being not mobile and the roots fixing the (typical) tree to the ground. Finally, a forest is commonsensically defined as a group of trees. The second concept, (traffic) sign, may come in many forms (as we know from own experience), but the first that comes to mind is the most typical one: the signpost. The signpost is typically fixed on the ground near a road, and has a post supporting a surface panel depicting some traffic-related information (labeled I_1 in the lower left corner of Fig. 8). The cognitive advantage of a signpost is that it has a recognisable physical structure, while “traffic sign” is so generic as to be a merely functional-based concept: any kind of surface panel depicting some traffic-related information is a traffic sign.

The generic space G of concept blending corresponds to the anti-unification shown as $G = I_1 \sqcap I_2$ in Fig. 8; G depicts common structure between a signpost and a tree: a stick-like object, fixed to the ground, and supporting another object on top. As discussed later, this common structure is the basis for a metaphor like “a forest of signs” to make sense—in contradistinction to a metaphor that does not make sense such as “a forest of chairs”, even when a typical chair is made of wood.

Now, the construction of the blended metaphor for sign forest can be interpreted easily in the combined generalisation-based analogy and amalgam framework: the input spaces can be generalised in different ways (although always satisfying what they already have in common, namely G). Different generalisations would yield different amalgams, but the one we are considering here can be seen as generalising I_2 into \bar{I}_2 , as shown in Fig. 8. Now this generalisation \bar{I}_2 can directly be unified with I_1 , since \bar{I}_1 is identical to I_1 ; this unification yields the amalgam $A = \bar{I}_1 \sqcup \bar{I}_2$ that, as shown in Fig. 8, represents a “forest of signposts”. Moreover, since $I_1 \equiv \bar{I}_1$, this model is an asymmetric amalgam, as evidenced by the fact that we generalise the source (Forest) until it unifies with the target (Signpost), while the latter remains fixed (i.e., is not generalised).

4.3. *Implementing the Model’s Mechanism(s) Using HDTP and Asymmetric Amalgams*

Besides the theoretical developments, one of the project aims of COINVENT was an implementation of the developed general model of concept blending 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 generalisation 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:

1. **Find joint generalisation G of input theories S and T :** Given two input domain theories S and T , the (set of) common generalisation(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 generalisation under restricted higher-order anti-unification is (still) not unique (also see Section 3.1), the anti-unification itself returns several possible least general generalisations out of which the system choses one generalisation G_x using heuristics.

2. **Reuse set of anti-unifications to get complete generalised source theory S' :** Given this generalised 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 generalise the previous source domain theory S as far as possible into the generalised 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. 7, and \models indicates the classical semantic consequence relation in the logical sense.)
3. **Compute assymetric amalgam between S' and T as proto-blend T' :** Now, given S' , we can compute the asymmetric amalgam T_a 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, due to the applied approach, 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 $\{a | a \in S' \wedge \phi_{x,S}(a) \notin S_c\}$ and from $T \setminus T_c$, respectively, have to be checked.⁵ Subsequently, the remaining axioms from both theories are added as additional elements to the resulting set of axioms T_a , resulting in an enriched target theory T' .⁶
4. **Fully instantiate proto-blend T' into blend T_B :** 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 generalisation 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. **Check for consistency of blend T_B , trigger repair if needed:** 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 “un-affected” axioms and (via generalisation 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

⁵The maximality of the outcome is rooted in HDTP’s previously mentioned coverage maximisation.

⁶Note that the unifications and addition of axioms conserve the \models relation between theories and, thus, the subsumption ordering as indicated in Fig. 7.

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 remains only a partial solution on the way to completely solving computational concept blending (even when already restricted to the case of logic-based domain theories), 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 subsections, we want to further exemplify our approach in two application cases: a concept blending account of Pegasus as combination between a horse and a bird in classical mythology, and a re-creation of the concept of a foldable toothbrush. While being structurally similar to each other, the application domains vastly differ—mythology and imaginative thought versus product development with high practical relevance—and give evidence of both, the pervasiveness of concept blending as cognitive phenomenon and the generality and domain-independence of the described approach.⁷ Additionally, besides exemplifying the functioning of the just described algorithm, the two examples (together with the reconstruction of the classical Rutherford analogy given by Besold, Kühnberger, and Plaza (2015)) thereby show that the newly conceived model for concept blending truly constitutes a conservative extension of the accounts of generalisation-based analogy and amalgamation integrated into it (logically resulting from the central roles assigned to both theories in the conceptual commitments in Section 2).

4.4. *Case Study 1: 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.

Contrary to the Rutherford analogy and several other examples from the cognitive AI literature, in the Pegasus case analogy-making alone would not be sufficient to model the genesis of the concept of a winged horse: Rutherford's achievement was to recognize the structural similarities between a schematic account of the solar system (the sun in the center, celestial bodies with lesser mass revolving around it) and his conception of the structure of the atom (the nucleus in the center, lightweight electrons in the surrounding), and to use the more precise understanding of the solar system to also sharpen his theory about the atom (introducing the idea of the revolution of the electrons along stable orbits

⁷The conceptual theories used in both examples for the sake of clarity have been reduced to what the authors think are the most basic characterisations of the involved concepts. Of course, for each of the conceptual spaces much richer formalisations would be possible, e.g., adding a beak to the characterisation of a bird, or a tail to the horse theory. Still, these additions would not impede the functioning of the proposed mechanism in a principled way, but instead would combinatorially increase the complexity of the reasoning process (for instance most likely resulting in a higher number of competing candidate theories in the generalisation step). Still, this complexity would be counteracted by the respectively corresponding selection heuristics assuring the effective functioning of the algorithm.

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 × clade, has_bodypart : entity × bodypart,
has_ability : entity × ability

Facts of the bird characterisation:
 (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)*

Facts of the horse characterisation:
 (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 4. Example formalisations of stereotypical characterisations for a bird S and a horse T .

Entities:
C : clade, E : entity

Facts:
 (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)*

Table 5. Abbreviated representation of the shared generalisation G based on the stereotypical characterisations for a horse and a bird, constituted by generalisations $\alpha_1 = \phi_S(\gamma_1)/\beta_1 = \phi_T(\gamma_1)$, $\alpha_2 = \phi_S(\gamma_2)/\beta_2 = \phi_T(\gamma_2)$, $\alpha_3 = \phi_S(\gamma_3)/\beta_3 = \phi_T(\gamma_3)$, and $\alpha_5 = \phi_S(\gamma_4)/\beta_4 = \phi_T(\gamma_4)$ (i.e., $S_c = \{\alpha_1, \alpha_2, \alpha_3, \alpha_5\}$ and $T_c = \{\beta_1, \beta_2, \beta_3, \beta_4\}$).

around the nucleus). Accordingly, computational accounts of the Rutherford analogy focus on transferring the governing laws from the (better informed) solar system domain to the (up to that point law-free) atom model domain. But compiling the Pegasus blend between conceptual theories for birds and horses requires the combination of qualitatively equally informed, with respect to the context fairly complete conceptual theories, most likely also involving inconsistency handling in the resulting blend (due to conflicting defining characteristics of the input domains). In this example we, thus, expect the model to produce a knowledge-rich but initially inconsistent conceptual theory as blend, forcing itself into inconsistency resolution before returning the final output.

We start with the conceptual theories (i.e., stereotypical characterisations) of a horse and a bird in a many-sorted first-order logic representation (as used by HDTP) from Table 4.

Given these characterisations, HDTP can be used for finding a common generalisation of both (Table 5),⁸ basically describing an entity of some clade, having a torso and legs, and being able to walk—a characterization covering (at least) most land-based vertebrates except for limbless or limb-reduced reptiles and amphibians.

Subsequently, the anti-unifications inversely corresponding to ϕ_S are re-used for generalising the entire source theory S (and not only S_c , i.e., the part covered by the original generalization G) into S' as given in Table 6. The resulting theory has the shared generalisation from Table 5 as a real subset, adding the abilities to fly and lay-eggs, as well as wings as part of the body, to the mix.

This conceptual theory S' of a, among others, winged, oviparous entity of some clade, is now used for computing the asymmetric amalgam with the (fixed) target theory T , consisting of descriptions of stereotypical features of a horse. As a result we obtain the

⁸As stated previously, when using HDTP the required subsumption relation between theories currently is given by logical semantic consequence \models , i.e., $A \sqsubseteq A'$ if $A' \models A$ for any two theories A and A' . In order to make sure that this relationship is preserved by HDTP's syntax-based operations, the range of admissible substitutions for restricted higher-order anti-unifications has to be further constrained to only allow for fixations and renamings.

Entities: C : clade, E : entity**Facts:**

(γ_1) *is_of.clade*(E, C) (γ_2) *has_bodypart*(E, \textit{legs})
 (γ_3) *has_bodypart*(E, \textit{torso}) (γ_4) *has_ability*(E, \textit{walk})
 (γ_5) *has_bodypart*(E, \textit{wings}) (γ_6) *has_ability*(E, \textit{fly})
 (γ_7) *has_ability*($E, \textit{lay_eggs}$)

Table 6. Abbreviated representation of the generalised source theory S' based on the stereotypical characterisations for a horse and a bird, including additional axioms γ_5 , γ_6 , and γ_7 obtained from generalizing the remaining axioms from $S \setminus S_c = \{\alpha_4, \alpha_6, \alpha_7\}$.

Entities: E : entity**Facts:**

(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, \textit{wings}) (δ_6) *has_ability*(E, \textit{fly})
 (δ_7) *has_ability*($E, \textit{lay_eggs}$)

Table 7. Abbreviated representation of the proto-blend T' obtained from computing the asymmetric amalgam between S' and T .

Facts:

(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*)

Table 8. Abbreviated representation of $T_B = \phi_T(T')$.

Facts:

(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*)

Table 9. Abbreviated representation of the final blended theory T_B giving a characterisation of pegasus after inconsistency check and repair (i.e., based on $S_{clash} = S \setminus \{\alpha_7\}$).

proto-blend T' from Table 7, featuring the axioms describing the horse (which, as part of the amalgamation process, have been unified with the corresponding axioms from S' , namely those giving the entity some clade, legs, a torso, and the ability to walk) and additionally the as yet unmatched axioms from S' assigning wings, and the abilities to fly and to lay eggs to some entity.

Therefore, as T' still features said axioms containing non-instantiated variables, the set of substitutions ϕ_T is applied to the theory resulting in the (with respect to ϕ_T) fully instantiated blend theory T_B from Table 8—the description of a mammal with legs, torso, and wings, being able to walk fly, and lay eggs.

In a concluding step, a consistency check of the blended theory T_B is performed. As already initially expected, taking into account world knowledge about mammals identifies a clash with the ability to lay eggs asserted in axiom δ_7 , as mammals generally are not oviparous (except for the subclass Prototheria as precisely defined special case, with no class member ever having been observed to have wings or be able to fly). Thus, returning to the start of the procedure, the algorithm is re-initiated, for example, with $S_{clash} = S \setminus \{\alpha_7\}$, and finally returns the (with respect to ϕ_T fully instantiated and consistent) version of T_B given in Table 9 as output: a mammal with torso, legs, and wings, being able to walk and fly.



Figure 9. A folding toothbrush like the one from the example in Section 4.5, characteristically featuring a hinge allowing the brush head to be folded back into the handle.

Sorts:
entity, part, functionality

Entities:
toothbrush, pocketknife : entity handle, brush_head, blade, hinge : part
brush, cut, fold : functionality

Predicates:
has_part : entity × part, has_functionality : entity × functionality

Facts of the pocketknife characterization:
 (α_1) *has_part(pocketknife, handle)* (α_2) *has_part(pocketknife, blade)*
 (α_3) *has_functionality(pocketknife, cut)* (α_4) *has_part(pocketknife, hinge)*
 (α_5) *has_functionality(pocketknife, fold)*

Facts of the toothbrush characterization:
 (β_1) *has_part(toothbrush, handle)* (β_2) *has_part(toothbrush, brush_head)*
 (β_3) *has_functionality(toothbrush, brush)*

Table 10. Example formalisations of stereotypical characterisations for a pocketknife *S* and a toothbrush *T*.

4.5. Case Study 2: The Folding Toothbrush

Folding toothbrushes are a conceptual combination between a typical stick-like toothbrush and a hinge-folding mechanism like that of a pocketknife (see Fig. 9).

As such, they represent a combination of two actual industry products subject to all the limitations and demands practicality and real-world applicability impose. In order to reconstruct the concept blending process at the heart of the invention of folding toothbrushes, analogous to the Pegasus case in the previous section, we start with the stereotypical characterisations of a standard toothbrush and a pocketknife in a many-sorted first-order logic representation from Table 10.

Given these characterisations, HDTP can be used for finding a common generalisation of both, for instance (due to the syntactic similarities and the system’s heuristics) aligning and generalising the axioms α_1 with β_1 (respectively asserting that the knife and the brush each have a handle), the blade α_2 with the brush head β_2 , and the functionality to cut things α_3 with the functionality to be used for brushing β_3 , resulting in some entity

Entities:

E : entity, P : part, F : functionality

Facts:

(γ_1) has_part(E , handle) (γ_2) has_part(E , P) (γ_3) has_functionality(E , F)

Table 11. Abbreviated representation of the shared generalisation G based on the stereotypical characterisations for a pocketknife and a toothbrush, constituted by generalisations $\alpha_1 = \phi_S(\gamma_1)/\beta_1 = \phi_T(\gamma_1)$, $\alpha_2 = \phi_S(\gamma_2)/\beta_2 = \phi_T(\gamma_2)$, and $\alpha_3 = \phi_S(\gamma_3)/\beta_3 = \phi_T(\gamma_3)$ (i.e., $S_c = \{\alpha_1, \alpha_2, \alpha_3\}$ and $T_c = \{\beta_1, \beta_2, \beta_3, \}$).

Entities:

E : entity, P : part, F : functionality

Facts:

(γ_1) has_part(E , handle) (γ_2) has_part(E , P)
 (γ_3) has_functionality(E , F) (γ_4) has_part(E , hinge)
 (γ_5) has_functionality(E , fold)

Table 12. Abbreviated representation of the generalised source theory S' based on the stereotypical characterisations for a toothbrush and a pocketknife, including additional axioms γ_4 and γ_5 obtained from generalizing the remaining axioms from $S \setminus S_c = \{\alpha_4, \alpha_5\}$.

Entities:

E : entity

Facts:

(δ_1) has_part(toothbrush, handle) (δ_2) has_part(toothbrush, brush_head)
 (δ_3) has_functionality(toothbrush, brush) (δ_4) has_part(E , hinge)
 (δ_5) has_functionality(E , fold)

Table 13. Abbreviated representation of the proto-blend T' obtained from computing the asymmetric amalgam between S' and T .

Facts:

(δ_1) has_part(toothbrush, handle) (δ_2) has_part(toothbrush, brush_head)
 (δ_3) has_functionality(toothbrush, brush) (δ_4) has_part(toothbrush, hinge)
 (δ_5) has_functionality(toothbrush, fold)

Table 14. Abbreviated representation of $T_B = \phi_T(T')$.

having a handle and (at least) one more part and a function (Table 11).

Subsequently, reusing the same anti-unifications applied in finding G (more precisely the ones inversely corresponding to the substitutions in ϕ_S), the source theory S is generalised into S' as given in Table 12: $\gamma_1, \gamma_2, \gamma_3$ form the joint generalization G , and the additional axioms γ_4 and γ_5 are obtained by generalising α_4 and α_5 , respectively. The resulting generalised source theory S' describes an entity with a handle, a hinge, and some additional part, which can be folded, and has some additional function.

Computing the asymmetric amalgam of this generalised source theory S' with the (fixed) target theory T (describing a toothbrush), we obtain the proto-blend T' from Table 13: the characterization of an object with a handle, a brush head, and a hinge, which can be used for brushing, together with an additional object describing some foldable entity.

As T' still features axiom δ_5 containing non-instantiated variables, the substitutions from ϕ_T (obtained in the original generalisation step from T to G) are applied to the theory resulting in the (with respect to ϕ_T) fully instantiated blend theory T_B from Table 14, describing an entity with a handle, a brush head, and a hinge, which can be folded and is usable for brushing: the concept of a hinge-equipped toothbrush that can be folded.

5. Conclusion: Related Work and Future Directions

In the previous sections we presented the analogy-inspired COINVENT model and a corresponding working algorithm for the blending of conceptual theories. The main contributions are the combination of generalisation-based analogy instantiated in (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 generalisation and domain matching asymmetric amalgams allow to soundly compute the concept blend of two input theories in a controlled fashion.

As should have become clear from the presentation of the model, and the discussion of its corresponding components, analogy plays a crucial role for COINVENT's take on concept blending. Still, there are also significant differences which may not be overlooked, with the distinct "power" of both mechanisms in our opinion being the most salient but also most important one. Analogy in most accounts is taken as a cognitive mechanism transferring knowledge from a better informed source domain into a sufficiently structurally similar target domain in a coherent fashion. Concept blending goes beyond this fairly conservative form of transfer in that the most general case does not recognize one domain as source and the other as target (with only the source contributing additional information to the target domain), but that both domains contribute information to the resulting blend, allowing for more combinatorial options—and consequently possible outcomes—than obtainable in the analogy setting.⁹ A dedicated discussion of the relationship between concept blending—and more precisely, COINVENT's account thereof—and analogy has, for instance, been provided by Besold (in press).

Also, ours is by far not the only current attempt at the computational modelling of concept blending. For instance Kutz, Bateman, Neuhaus, Mossakowski, and Bhatt (2015) give an account and a system model for the computer-based blending of ontologies (i.e., the conceptual theories involved as inputs for blending are ontological descriptions). The ontology-based approach views a concept as an ontological specification: a specification that is ideally so general as to cover all possible instances or occurrences of the concept. As such, certain properties and relations are selected to form these specifications that are useful for an ontology framework. In contrast, in its conceptual layout our approach is inspired by an idea also underlying Rosch (1988)'s prototype theory, namely the notion that concepts in human cognition can be characterized up to a high level of precision by bundles of their most typical properties (albeit typicality may certainly be context-dependent). This view is also taken in examples by Fauconnier and Turner (1998) that are used to show how conceptual blend works: a boathouse has typical properties of boat and house—but not other properties that may appear in an ontological specification of boat and house.

Martinez et al. (2014) presented an approach for the algorithmic blending of mathematical theories, trying to model the combination of previously independent mathematical concepts as basis for mathematical concept invention. Contrary to our setting, dealing with mathematical conceptual theories limits the range to unambiguous and logically unique axiomatisations, also removing the need for "semantic" consistency checks of the resulting blends within output theories or against world knowledge. The approach taken by Martinez et al. (2014) is similar to ours in that it also builds upon J. Goguen (2006)'s ideas and uses HDTP for finding shared generalisations between mathematical input

⁹Recall that the choice for asymmetric amalgams in the presentation of the COINVENT model had been motivated by considerations of simplicity. As explained in the corresponding footnote in Section 4.1, the required expansion of the model (and subsequently also the algorithm) is straightforward and basically consist of a doubling of already existing structures.

theories. Still, the blending mechanism works differently in that it basically relies on a simple generate-and-test approach to step-wise building up increasingly complex logically consistent combinations of the axioms from the input theories.

Compared to the earlier work by Martinez et al. (2012) on blending as general cognitive mechanism, our addition of amalgamation as formal description and guiding framework and the integration with the generalisation-based analogy mechanism goes beyond the work reported there. Moreover, Martinez et al. (2012) do not provide an algorithmic account but restrict themselves to outlining a fairly general computational-level description of the envisioned mechanism.

Another recent report of work on concept blending was described by Li, Zook, Davis, and Riedl (2012), who provide case studies of systems taking into account goals and contexts in the blending-based production of creative artefacts. They emphasise the context-dependent aspects of concept blending, moving focus away from the attributes of the input domains to the role the context plays in blend generation. In their model, the situational relevance and the communicative goal of a blend as metaphor-like speech element determine the blending process and outcome to a huge extent. While Li et al. (2012)'s work is similar to our general account in its emphasis on (cognitive representations of) real-world concepts and a certain influence from computational analogy-making in the described systems, their approach differs in that it relies on selective projections from the input spaces into the blend space without generalisation playing a prominent role. Also, again a clear description of the systems applied in the presented examples is lacking.

Based on this short comparison we think that the COINVENT model and system constitutes a valuable contribution to the ongoing discourse and a big step towards a computationally feasible model of concept blending. Its virtues lie, among others, in combining the generality of modelling introduced by HDTP's use of many-sorted first-order logic languages with the formal soundness and solid theoretical foundations of the underlying generalisation model and the amalgam framework, offering a computational- and algorithmic-level account for blending general conceptual theories.

Concerning the next steps of development of our concept blending framework, on the formal side the restriction on the substitutions used in HDTP has to be weakened and finally removed in order to access the frameworks full generality and expressivity (i.e., 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 as the former does not hold anymore between successive generalisation steps as soon as permutations or argument insertions are applied. Here, we hope that providing a semantics to the syntax-based operations in HDTP and restricted higher-order anti-unification—for instance by an approach similar to the derived signature morphisms discussed by Mossakowski, Krumnack, and Maibaum (2015)—will allow us to subsequently construct a suitable substitute.

A more system-oriented open question is the further integration of heuristic and knowledge-sensitive methods during blend computation and selection for modelling contextual constraints or internal properties of the cognitive agent (as also described in the overview of conceptual commitments in Section 2). Heuristics can be applied at different points of the proposed concept blending: they form essential part of HDTP's computation of the least general generalisation between the input domains and also promise to make the inconsistency resolution as final step before outputting a blended theory more efficient. Still, they can also be used for modelling a cogniser's expertise or similar individual features. For instance, already during the computation of generalisations cer-

tain combinations of elements from the respective input domains could be favoured over others, and during inconsistency resolution focus could not only be put on efficiency in finding a consistent conceptual theory but content-related aspects could be taken into account. Moreover, also the integration of background knowledge about the environment or the task as guiding forces for the blending process and its output clearly would be desirable. While this seems fairly straightforward during inconsistency resolution, whether and how to already inform the earlier steps of the algorithm remains an open question.

Finally, resonating with the corresponding remark at the end of Section 4.1, a more general challenge not only relevant for our system but for a significant part of computational creativity as a research discipline concerns theories and computationally feasible methods for evaluating the usefulness and the novelty of the output of a computational system from either an agent-centric or a general perspective. While our current approach (i.e., simply equating usefulness with consistency) falls short of the requirements in most application scenarios or contexts, alternative proposals are either highly specific to a domain or task, or make recourse to external means of evaluation shifting the burden away from the cognitive system. As can be expected, focusing on the degree of novelty instead of usefulness as dimension further complicates the question.

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(1), 39–59.
- Besold, T. R. (in press). The relation of conceptual blendign and analogical reasoning. In R. Confalonieri et al. (Eds.), *Concept invention: Foundations, implementation, social aspects and applications*. Springer. (Forthcoming in 2017.)
- Besold, T. R., Kühnberger, K.-U., & Plaza, E. (2015). Analogy, amalgams, and concept blending. In *Proceedings of the third annual conference on advances in cognitive systems (poster collection)*. Cogsys.org.
- Besold, T. R., & Plaza, E. (2015). Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *Proceedings of the sixth international conference on computational creativity (iccc 2015)* (pp. 150–157). Provo, Utah: Brigham Young University.
- Boden, M. (1977). *Artificial intelligence and natural man*. Harvester Press.
- Boden, M. A. (2003). *The creative mind: Myths and mechanisms*. Routledge.
- Bou, F., Schorlemmer, M., Corneli, J., Gomez Ramirez, D., Maclean, E., Smaill, A., & Pease, A. (2015). The role of blending in mathematical invention. In *Proceedings of the sixth international conference on computational creativity (iccc 2015)* (pp. 55–62). Park City, Utah: Brigham Young University.
- Confalonieri, R., Corneli, J., Pease, A., Plaza, E., & Schorlemmer, M. (2015). Using argumentation to evaluate concept blends in combinatorial creativity. In *Proceedings of the sixth international conference on computational creativity (iccc 2015)* (pp. 174–181). Park City, Utah: Brigham Young University.

- Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1), 1 - 63. doi:
- Fauconnier, G., & Turner, M. (1998). Conceptual integration networks. *Cognitive Science*, 22(2), 133–187.
- Gentner, D., & Smith, L. A. (2013). Analogical learning and reasoning. In D. Reisberg (Ed.), *The Oxford Handbook of Cognitive Psychology* (pp. 668–681). Oxford University Press.
- Goguen, J. (2006). Mathematical models of cognitive space and time. In *Reasoning and cognition; proceedings 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* (p. 291-316). Springer.
- Kutz, O., Bateman, J., Neuhaus, F., Mossakowski, T., & Bhatt, M. (2015). E pluribus unum. In T. R. Besold, M. Schorlemmer, & A. Smail (Eds.), *Computational creativity research: Towards creative machines* (Vol. 7, pp. 167–196). Atlantis Press.
- Kutz, O., Mossakowski, T., Hois, J., Bhatt, M., & Bateman, J. (2012). Ontological Blending in DOL. In *Proceedings of the 1st International Workshop on "Computational Creativity, Concept Invention, and General Intelligence"*.
- Li, B., Zook, A., Davis, N., & Riedl, M. (2012). Goal-driven conceptual blending: A computational approach for creativity. In *Proceedings of the Third International Conference on Computational Creativity* (pp. 9–16).
- Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. New York, NY, USA: Henry Holt and Co., Inc.
- Martinez, M., Besold, T. R., Abdel-Fattah, A., Gust, H., Schmidt, M., Krumnack, U., & Kühnberger, K.-U. (2012). Theory Blending as a Framework for Creativity in Systems for General Intelligence. In P. Wang & B. Goertzel (Eds.), *Theoretical Foundations of Artificial General Intelligence* (pp. 219–239). Atlantis Press.
- Martinez, M., Krumnack, U., Smail, A., Besold, T. R., Abdel-Fattah, A. M., Schmidt, 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, pp. 180–192). Springer.
- Mossakowski, T., Krumnack, U., & Maibaum, T. (2015). What is a derived signature morphism? In M. Codescu, R. Diaconescu, & I. Tutu (Eds.), *Recent trends in algebraic development techniques: 22nd international workshop, wadt 2014, sinaia, romania, september 4-7, 2014, revised selected papers* (pp. 90–109). 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, pp. 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, pp. 312–326). Springer.
- Pereira, F. C. (2007). *Creativity and AI: A conceptual blending approach*. Mouton de Gruyter.
- Pereira, F. C., & Cardoso, A. (2003). Optimality principles for conceptual blending: A first computational approach. *AISB Journal*, 1(4).
- Plotkin, G. D. (1970). A note on inductive generalization. *Machine Intelligence*, 5, 153–163.
- Rosch, E. (1988). Principles of categorization. In A. Collins & E. E. Smith (Eds.), *Read-*

- ings in cognitive science, a perspective from psychology and artificial intelligence* (pp. 312–322). Morgan Kaufmann Publishers.
- 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* (pp. 163–194). Springer.
- Schorlemmer, M., Confalonieri, R., & Plaza, E. (2016). Coherent concept invention. In *Proceedings of the Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI 2016)*. CEUR-WS.org.
- Schwering, A., Krumnack, U., Kühnberger, K.-U., & Gust, H. (2009). Syntactic principles of Heuristic-Driven Theory Projection. *Journal of Cognitive Systems Research*, 10(3), 251–269.
- Thagard, P., & Stewart, T. C. (2011). The aha! experience: Creativity through emergent binding in neural networks. *Cognitive Science*, 35(1), 1–33.
- Veale, T., & O'Donoghue, D. (2000). Computation and blending. *Cognitive Linguistics*, 11(3/4), 253–281.