Theory blending: extended algorithmic aspects and examples

Annals of Mathematics and Artificial Intelligence manuscript No.
(will be inserted by the editor)

Algorithmic Aspects of Theory Blending

Maricarmen Martinez · Ulf Krumnack · Alan Smaill · Tarek Besold · Ahmed Abdel-Fattah · Martin Schmidt · Helmar Gust · Kai-Uwe Kühnberger · Markus Guhe · Alison Pease

Received: date / Accepted: date

Abstract In Cognitive Science, conceptual blending has been proposed as an important cognitive mechanism that facilitates the creation of new concepts and ideas by constrained combination of available knowledge. It thereby provides a possible theoretical foundation for modeling high-level cognitive faculties such as the ability to understand, learn, and create new concepts and theories. Quite often the development of new mathematical theories and results is based on the combination of previously independent concepts, potentially even originating from distinct subareas of mathematics. Conceptual blending promises to offer a framework for modeling and re-creating this form of mathematical concept invention with computational means. This paper describes a logic-based framework which allows a formal treatment of theory blending (a subform of the general notion of conceptual blending with high relevance for applications in mathematics), discusses algorithmic aspects of blending within the framework, and provides an illustrating worked out example from mathematics.

Keywords Concept Blending · HDTP

Mathematics Subject Classification (2000) 03B

Maricarmen Martinez
Universidad de los Andes, Bogotá, Colombia E-mail: m.martinez@uniandes.edu.co

Ulf Krumnack · Tarek Besold · Martin Schmidt · Helmar Gust · Kai-Uwe Kühnberger
Institute of Cognitive Science, University of Osnabrück, Germany
E-mail: {krumnack,besold,hgust,kkuenbe}@uos.de

Alan Smaill · Markus Guhe
School of Informatics, University of Edinburgh, Scotland E-mail: {A.Smaill,m.guhe}@ed.ac.uk

Ahmed Abdel-Fattah
Department of Mathematics, Faculty of Science, Ain Shams University, Cairo
E-mail: ahabdelfattah@sci.asu.edu.eg

Alison Pease
School of Computing, University of Dundee, Scotland E-mail: a.pease@dundee.ac.uk

1 Introduction

Conceptual blending theory (CB) [8] provides a mechanism by which novel ideas and meanings are produced by combining familiar ideas in an unfamiliar way. For instance, "trashcan basketball" integrates knowledge structures from trash disposal and conventional basketball to yield a blend: this is comprised of structure from each of the two domains as well as unique structure of its own [6]. The theory has gained popularity as a way of explaining high-level cognitive and linguistic phenomena, such as metaphor, analogy, metonymy and counterfactual reasoning [14, 1]. Even if only a few of the assumptions made about the importance of blending mechanisms within human cognition and intelligence turn out to be correct, a complete and implementable formalization of CB and its defining characteristics would promise to trigger significant development in artificial intelligence and any other field aiming at modeling or re-implementing capacities related to human intelligence with computational means. The original account of CB in [8], however, lacks a formal or algorithmic account, and, to date, no such account has been forthcoming.

CB is also considered to play a crucial role in mathematical invention and theory development. Lakoff and Núñez [13] present a blending-based account of the origin and development of mathematical ideas is presented, in which human mathematics is grounded in the bodily experience of physical interactions in the world and an inheritance or transfer process of these experiences to the domain of mathematical concepts. In this account, humans start out with very simple notions and subsequently, by successive combination of concepts, over time develop these into more and more complex theories giving rise to the whole of mathematics as a discipline and academic field of research (also see [2]). While the original account from [13] has been criticized and further developed by other researchers over the last 15 years (see, e.g., [21] for a reply and further development of the ideas from [13]), the basic intuition of complex, abstract concepts arising from iterated combinations of simpler, more grounded ones still holds and by now is regarded as largely uncontroversial. Based on this, CB promises to offer a theoretical framework within which to further study and (if possible) computationally re-implement the corresponding cognitive processes.

When considering CB in mathematics, due to the axiomatized nature of mathematics, the most relevant form of blending is the combination of theories (as opposed to, e.g., multimodal blending of concepts and sensory modalities in arts or the blending of vague linguistic concepts). Mathematical concepts can be defined by finitely axiomatized theories in a logic, and combining concepts means the combination of two concept axiomatizations. This form of concept blending will consequently be referred to as theory blending.

This paper is structured as follows: in the remainder of the introduction we will briefly survey computational approaches to CB and give a short overview of an (unfortunately unfinished) formal account of CB developed by Goguen and our related overall approach to theory blending. In the following section we introduce the formal framework that we use to model blending processes. In Section 3, we elaborate our proposal for an algorithmic description of theory blending. We discuss what happens when our procedure enters into "relaxation" stages and offer some considerations of efficiency in Section 4. As a proof of concept, we illustrate our algorithm with a worked example in Section 5, and finally present our concluding remarks, review of related work, and an outlook for future research in the last section.

1.1 Computational Accounts of Concept Blending

CB is a complex and powerful theory, and there are no fully implemented accounts as yet. However, advances have been made along multiple dimensions.

The earliest computational models of concept blending, [25] and [19], were based on Gentner's structure-mapping theory (SMT) of analogy [9]. The former used semantic network representations of domains, and the latter genetic algorithms to search the space of possible blends. Both, however, relied on handcrafted knowledge: a common issue in CB models. Besold et al. [3] and [4] show how work on computational analogy models which use generalization followed by mapping (such as HDTP), and amalgamation (combining solutions from multiple cases in CBR), as opposed to SMT, can be used in blending. Other key advances include determining the fundamental characteristics of a good blend: for instance, Martins et al.[17] investigated criteria for creative concept-blends, by asking participants to rate human-generated concept blends in terms of some of the optimality principles proposed by [8] and other principles connected to creativity. Confalonieri et al. provide an alternative take on the problem [5], proposing to use computational argumentation for evaluating concept blends; through an open-ended and dynamic discussion, through which meaning is constructed and blends are refined and improved. In a similar social context, Li et al. [15] provide a computational perspective to the notion that blending theory must take communication contexts and goals into consideration. That is, a blend may have a plurality of meanings, and can only be properly understood within the context in which they arise. Li et al. use these concepts to clarify, constrain and implement computational procedures which are ambiguous in the original non-computational theory. Many models are open to the criticism that the input conceptual spaces consist of handcrafted knowledge: in [24], Veale offers an alternative by introducing the notion of a conceptual mash-up, a form of blending which uses a technique Veale calls "google-milking". This uses common questions on the web to find salient properties of a concept, which are then used to drive the blend. This follows up previous work by Veale, [23], in which he developed a CB model which automatically found its input spaces from Wikipedia and Wordnet, and used blending theory to understand novel portmanteau words such as "Feminazi" (Feminist + Nazi). Xiao and Linkola [26] have investigated blending in the context of different forms of spaces and blends: their model of multi-media blending – Vismantic – takes in a subject and message, such as "electricity is green", finds images for each word on flickr, and applies juxtaposition, fusion and replacement to the photos found, outputting a image which blends the two concepts. The question of what sort of spaces can be blended is considered by Kutz et al. [11], who investigate the principles of blending at the level of ontologies, and show how the Ontohub/Hets ecosystem can be used to support the generation and evalution of ontological blendoids.

We base our formal model, elaborated below, on Goguen's logic-based approach.

$$\begin{tikzcd}
& G \arrow[dl] \arrow[dr] \arrow[dd] & \\
I_1 \arrow[dr] & & I_2 \arrow[dl] \\
& B &
\end{tikzcd}$$

Fig. 1 Goguen's version of concept blending (cf. [10]).

1.2 Goguen's Account of CB and Our Overall Approach

An early formal account on CB, especially influential to our approach, is the classical work by Goguen using notions from algebraic specification and category theory [10]. This version of CB is depicted in Figure 1, where a blend of two inputs I_1 and I_2 is shown. Each node in the figure stands for a representation of a concept or conceptual domain as a theory, i.e., as a finite set of axioms in a formal language. We will call the nodes "spaces", so as to avoid terms with strong semantical load such as "concept" or "conceptual domain". Each arrow in the figure stands for a morphism, that is, a change-of-language *partial* function that translates at least part of the axioms from its domain into axioms in its codomain, preserving their structure. Now, while in practice all formal languages of interest have an established semantics and the morphisms are therefore intended to act as partial interpretations of one theory into another, Goguen's presentation of CB stays at the syntactic level, which more directly lends itself to computational treatment. The same will apply to our own approach. Given input spaces I_1 and I_2 and a generalization space G that encodes some (ideally all) of the structural commonalities of I_1 and I_2, a blend diagram is completed by a blend space B and morphisms from I_1 and I_2 to B such that the diagram (weakly!) commutes. This means that if two parts of I_1 and I_2 are translated into B and in addition are identified as 'common' by G, then they must be translated into exactly the same part of B (whence the term 'blend').

A standard example of CB, discussed in [10] and linked to earlier work on computational aspects of blending in cognitive linguistics (see, e.g., [25]), is that of the possible blends of HOUSE and BOAT into both BOATHOUSE and HOUSEBOAT (as well as other less-obvious blends). Parts of the spaces of HOUSE and BOAT can be structurally aligned (e.g. a RESIDENT LIVES-IN a HOUSE; a PASSENGER RIDES-ON a BOAT). Conceptual blends are created by combining features from the two spaces, while respecting the constructed alignments between them. Newly created blend spaces are supposed to coexist with the original spaces: we still want to maintain the spaces of HOUSE and BOAT.

A still unsolved question is to find criteria to establish whether a certain blend is better than other candidate blends. This question has lead to the formulation of various competing optimality principles in cognitive linguistics (cf. [8]). While several of them involve semantic aspects that escape Goguen's and our own treatment of CB, other principles can be reasonably approached even from a more syntactic framework. For example, there is the *Web Principle* (maintain as tight connections as possible between the inputs and the blend), the *Unpacking Principle* (one should be able to reconstruct the inputs as much as possible, given the blend), and the *Topology Principle* (the components of the blend should have similar relations to those that their counterparts hold in the input spaces). These three principles, taken as a package, can be interpreted in terms of Figure 1 as demanding that the morphisms should preserve as much representational structure as possible. For example, one can notice that Figure 1 looks like the diagram of a pushout in category theory. Goguen actually argued against forcing the diagram of every blend to be a pushout [10], but he did claim that some forms of a pushout construction (in a $\frac{3}{2}$-category) capture a notion of *structural optimality* for blends.

We will propose two alternative competing criteria for structural blend optimality that also work in the spirit of the Web, Unpacking, and Topology principles, and an algorithmic method for performing blending guided by those principles. We will use HDTP, a framework for computational analogy making between many-sorted *first-order* theories, in order to obtain the generalization spaces G. Accordingly, our presentation in the following will be restricted to CB over first-order theories.

2 Our Framework

According to Figure 1, the task of finding a blend diagram, given two inputs, requires finding a generalization G, a blend space B, and the arrows of the diagram. In this section we present our approach to this problem. As it will be clear, we will use previous work on analogy-making in order to find G, so our new contribution will focus on the issue of finding B, given two input theories and a generalization G.

2.1 Generalization Finding

Our approach is based on *Heuristic-Driven Theory Projection* (HDTP), which is a framework for computing analogical relations between two input spaces presented as axiomatizations in (possibly distinct) many-sorted first-order languages [22]. HDTP proceeds in two phases (Figure 2): in the *mapping phase*, the source and target spaces are compared to find structural commonalities and a generalized space, G, is created, which subsumes the matching parts of both spaces. In the *transfer phase*, unmatched knowledge in the source space can be transferred to the target space to establish new hypotheses. Our blending approach only needs the mapping phase of HDTP; the transfer phase will be replaced by a new blending algorithm in which the two inputs play a symmetric role. Accordingly, instead of talking about source and target spaces, from now on we will refer to the input spaces simply as L and R, as mnemonics for "left" and "right" in our graphical depictions of blend diagrams, but without implying any asymmetry in the role of input spaces.

$$\begin{tikzcd} & \textit{Generalization (G)} \arrow[dl] \arrow[dr] & \\ \textit{Source (L)} \arrow[rr, dotted, "\textit{analogical transfer}"] & & \textit{Target (R)} \end{tikzcd}$$

Fig. 2 HDTP's overall approach to creating analogies (cf. [22]).

During the mapping phase in HDTP, pairs of formulae from L and R are *anti-unified*, resulting in a generalization theory G that reflects common aspects of the input spaces. Anti-unification [20] is a mechanism that finds least-general anti-unifiers of expressions (formulae or terms). An anti-unifier of A and B is an expression E such that A and B can be obtained from E via substitutions. E is a least-general anti-unifier of A and B if it is an anti-unifier such that the only substitutions on E that yield anti-unifiers of A and B act as trivial renamings of the variables in E. First-order anti-unification, where only first-order substitutions are allowed, is not powerful enough to capture structural commonalities and produce the generalizations needed in HDTP. A special form of *higher-order* anti-unification is therefore used where, under certain conditions, symbols of relation and function can also be included in the domain of substitutions (see [22] for the details). The generalized theory G can be projected into the original spaces by higher-order substitutions which are computed by HDTP during anti-unification. We will say that a formula is *covered* by G if it is in the image of this projection; otherwise it is *uncovered*. Two formulae (or terms) from the input spaces that are generalized (i.e. anti-unified) to the same expression in G are considered to be analogical. In analogy making, the analogical relations are used in the transfer phase to translate uncovered facts from the source to the target space, while blending combines uncovered facts from both spaces. The blending process can thus build on the generalization and substitutions provided by the analogy engine, and analogy can be considered a special case of blending.

Axiomatization L		**Axiomatization** R	
$x \leq_L x$	(L1)	$x \leq_R x$	(R1)
$x \leq_L y \wedge y \leq_L z \rightarrow x \leq_L z$	(L2)	$x \leq_R y \wedge y \leq_R z \rightarrow x \leq_R z$	(R2)
$x \leq_L y \vee y \leq_L x$	(L3)	$x \leq_R y \vee y \leq_R x$	(R3)
$1 \leq_L x$	(L4)	$0 \leq_R x$	(R4)
$x +_L y = y +_L x$	(L5)	$x +_R y = y +_R x$	(R5)
$(x +_L y) +_L z = x +_L (y +_L z)$	(L6)	$(x +_R y) +_R z = x +_R (y +_R z)$	(R6)
$\neg(x +_L 1 \leq_L x)$	(L7)	$x +_R 0 = x$	(R7)
$x \leq_L y \wedge y \leq_L x +_L 1 \rightarrow y = x \vee y = x +_L 1$	(L8)	$x <_R y \rightarrow \exists z : x <_R z \wedge z <_R y$	(R8)

Generalization G	
$x \leq x$	(G1)
$x \leq y \wedge y \leq z \rightarrow x \leq z$	(G2)
$x \leq y \vee y \leq x$	(G3)
$a \leq x$	(G4)
$x + y = y + x$	(G5)
$(x + y) + z = x + (y + z)$	(G6)

Fig. 3 The two axiomatizations and the first generalization G used in the worked example. G comes together with a **left substitution** $\{a \mapsto 1, \leq \mapsto \leq_L, + \mapsto +_L\}$ and a **right substitution** $\{a \mapsto 0, \leq \mapsto \leq_R, + \mapsto +_R\}$ from which L and R can be recovered.

$$\begin{tikzcd} & G \arrow[dl, "\cong"'] \arrow[dr, "\cong"] & \\ L \arrow[rr, dotted, "\cong"] \arrow[dr, "\cong"'] & & R \arrow[dl, "\cong"] \\ & L \cong R & \end{tikzcd} \qquad \begin{tikzcd} & \emptyset \arrow[dl] \arrow[dr] & \\ L \arrow[dr] & & R \arrow[dl] \\ & L \oplus R & \end{tikzcd}$$

Fig. 4 The two extreme cases of input spaces, along with their generalizations and blends.

Example 1. We will use a working example in this paper based on the theories L and R from Table **??**, which describe basic properties of the standard order and addition of the natural numbers (starting from 1) and the non-negative rationals, respectively. All the axioms are implicitly universally quantified, and $x <_S y$ abbreviates $\neg(y \leq_S x)$, for $S \in \{L, R\}$. The table also shows a generalization theory G over the signature $\{a, \leq, +\}$, which reflects the fact that axiom (Li) is structurally like (Ri) when $1 \leq i \leq 6$. Upon applying the left and right substitutions to G, we will get the first six L-axioms and the first six R-axioms, respectively, which are the covered formulae in this example.

2.2 Optimal Blends

There are two extreme cases of CB, depending on the portion of the input theories covered by G. The first case (left side of Figure 4) occurs when the input spaces are isomorphic, meaning that there is a bijective morphism that simply renames the signature symbols of the language of L onto the symbols of R. In that case, all formulae of the theories can be generalized and are completely covered by G, and the resulting blend will be isomorphic to both of them. The other extreme (right side of Figure 4) occurs when no formulae can be aligned and therefore the generalized theory G is empty, so no formulae of the input theories are covered. In this case, a blend can always be obtained by taking the (possibly inconsistent) disjoint union of the input theories.[1] In practice, neither of the two extreme cases is of a real interest. The interesting *proper blends* arise when only parts of the input theories are covered by G. In fact, one can adjust the blend by changing the generalization, either by removing formulae from G and so reducing its coverage, or by choosing altogether another G which associates different formulae.

Given the generalization G, the theories L and R can be split into their (non-empty) covered parts L^+_G and R^+_G and uncovered parts L^-_G and R^-_G. The covered parts are fully analogical, i.e. basically isomorphic, and make up the core of a blend B based on G. The uncovered parts reflect the idiosyncratic aspects of the spaces, which we would ideally want to integrate into B. However, due to the identifications induced by G, adding all this to B may result in an inconsistent theory. To preserve consistency, we may be forced to consider only consistent subsets of this ideal, fully inclusive, blend. In view of this, we propose to define optimality of blends (see Definition 1) using the following two optimality principles:

[1] HDTP is syntax-based, but has some "re-representation" abilities by which formulae derived from the axioms may be used in the mapping phase if the original axiomatizations do not yield a good analogical relation (cf. [22, pp. 258]). Thus, in some cases, two formally different but semantically equivalent axiomatizations may not result in an empty generalization.

Compression Principle (CP) aim for blend diagrams in which B is as compressed as possible, that is, where as many signature symbols aligned by G as possible are actually integrated as a single symbol in B.

Informativeness Principle (IP) aim for blend diagrams in which B is as informative as possible, i.e., it includes a maximally consistent subset of the potentially merged formulae (obtained by taking the union of the input theories and then collapsing pairs of signature symbols that have been identified by the analogy into one unified symbol).

Note that IP renders a version of the Web and Topology principles formulated in the introduction, while CP supports the Unpacking Principle.

Definition 1. We call a blend diagram *optimal* if its blend space is consistent and satisfies CP and IP. That is, if it is consistent and as maximally compressed and informative as possible.

2.3 Searching for Optimal Blends

Just as Figure 2 and Table **??** suggest, every generalization we use, say H, will come in association with both a partial signature morphism λ_H from the signature of H to Σ_L and a partial signature morphism ρ_H from the signature of H to Σ_R. We will use the notation $\mathbf{H} = \langle H, \lambda_H, \rho_H \rangle$ whenever we need to encode this full structure, and we will say that $\mathbf{H}$ is a *relaxation* of $\mathbf{G} = \langle G, \lambda_G, \rho_G \rangle$ if $H \subseteq G$, $\lambda_H \subseteq \lambda_G$, and $\rho_H \subseteq \rho_G$.

$$
\begin{tikzcd}
 & H \arrow[dl] \arrow[dr] \arrow[dd] & \\
L \arrow[dr] & & R \arrow[dl] \\
 & B &
\end{tikzcd}
$$

Fig. 5 An element of our search space.

With that in mind, we can now state the problem we want to solve. We take as given two first-order theories L and R over signature Σ_L and Σ_R, respectively, and a generalization $\mathbf{G}$ of these two theories (we have in mind a generalization found by HDTP to be as good as possible in terms of coverage). We want to find, in an algorithmic way, all the optimal blend diagrams of the form shown in Figure 5 that satisfy all of the following constraints:

1. $\mathbf{H}$ is a *relaxation* of $\mathbf{G}$.
2. The signature Σ_B of B is a '*right collapsed union*' of Σ_L and Σ_R constructed thus: add to Σ_B all the uncovered symbols from both input signatures, and, in addition, for each pair of symbols $s_L \in \Sigma_L$ and $s_R \in \Sigma_R$ that are aligned by the generalization $\mathbf{H}$, add the symbol from Σ_R to Σ_B. In the last case, we say that the two symbols were *collapsed* into one.
3. The covered part of L, L_H^+, must be a subset of B.
4. Every formula in B that is not in L_H^+ must belong to $Ax_H = Tr_H(L_H^-) \cup R_H^-$, where $Tr_H(L_H^-)$ is obtained from L_H^- by replacing any symbol of Σ_L (covered by H) by its counterpart in Σ_R. This ensures that all formulae of Ax_H are built over the signature Σ_B.

Notice that applying condition (2) above to the theories of Example 1, yields that Σ_B will coincide with Σ_R, since no symbol in Σ_L is uncovered by the left substitution. We may need to find an example where at least one more symbol is uncovered in one of the spaces L and R.

It is tempting to conclude, also from condition (2) above, that our approach is biased towards one of the two input domains, as it always prefers choosing vocabulary from the left input space when forming blends. However, as it will be clear later, the core of our algorithmic approach is unchanged if a different symbol collpasing method is used to form the signature Σ_B. Alternatively, we could extend our algorithm with a final step that produces, for each discovered optimal blend, all of its "mirror" blends, obtained by alternative choices of vocabulary. This is the reason why we claim the treatment of the two input spaces is essentially symmetric.

With one more piece of notation that will also be useful later, we will be able to reformulate our search problem in a more concise way. Let $\mathbf{B} = \langle H, \lambda_H, \rho_H, B\rangle$ denote a blend diagram such as that of Figure 5. This notation does not explicitly include all the morphisms of the diagram, but only those from the generalization to the inputs, since all others are trivial to fill-in if needed (they are partial identity functions betweeen signatures or translations using the Tr_H). Then, we want an algorithm that, given L, R and $\mathbf{G}$, will explore (in search of all the optimal blends) the space of all blend diagrams of the form $\mathbf{B} = \langle H, \lambda_H, \rho_H, B\rangle$ for which the two following conditions hold:

1. $\mathbf{H} = \langle H, \lambda_H, \rho_H\rangle$ is a *relaxation* of the generalization $\mathbf{G} = \langle G, \lambda_G, \rho_G\rangle$, in the sense that $\mathbf{H}$ can be obtained from $\mathbf{G}$ by dropping one or more of the renamings of symbols induced by $\mathbf{G}$, so that $H \subseteq G$, $\lambda_H \subseteq \lambda_G$, $\rho_H \subseteq \rho_G$.
2. $R_H^+ \subseteq B \subseteq R_H^+ \cup Ax_H$.

The above conditions can be summarized in plain language by saying that the search space (given the fixed optimal generalization G provided by HDTP) is the collection of all blend diagrams that are at least as informative as some $\langle H, \lambda_H, \rho_H, R_H^+\rangle$, where $\langle H, \lambda_H, \rho_H\rangle$ is a relaxation of $\mathbf{G}$. Making H larger means moving in the search space towards more compressed blends, while letting H unchanged and enlarging B means moving towards more informative blends.

An unconstrained way to algorithmically identify a list of optimal blends leads to an explosion of possibilities to be tried, so good heuristics are needed in order to choose which possibilities to test first (see also Section 4). Notice that for a given generalization H, the formulae in Ax_H would give rise to $2^{|Ax_H|}$ possible ways in which a subset of zero or more of the $|Ax_H|$ unpaired formulae from both L and R can be formed (and thus a way in which a blend diagram in our search space, with generalization space H, can be formed). Extending a generalization H with each of these subsets results in $2^{|Ax_H|}$ corresponding sets that eventually form a network of theories isomorphic to the power set algebra of a set with $|Ax_H|$ elements. This network can thus be represented by a lattice $\mathcal{L}_{\mathcal{B}_H} = \langle \mathcal{B}_H, \subseteq\rangle$, where $\mathcal{B}_H$ is the set of all *potential blends* (based on Ax_H) and $\subseteq$ is the subset inclusion relation.

3 Theory Blending Algorithm

3.1 Overall Search Strategy

Given two input theories L and R over first-order signatures Σ_L and Σ_R, respectively, we propose to proceed according to the following general algorithm to find optimal blends. Figure 6 depicts an overall logical flowchart of the steps explained in the following.

1. Generalization Using the HDTP mapping phase, compute a generalization G that is as strong as possible (i.e., identifies as many symbols as possible) together with its associated substitutions[2]. As an example, see Table **??** and Example 1.
2. Identification Based on the current generalization $H \subseteq G$ (initially set to G), build a blend signature Σ_B by forming the 'right collapsed union' of Σ_L and Σ_R described in the previous section.
3. Blending Construct the set of all formulae over Σ_B that might be part of a blend. For a generalization $H \subseteq G$, this will consist of every formula in R_H^+ (the covered part of R) plus every formula in the uncovered parts of R and L (i.e., $Ax_H = Tr_H(L_H^-) \cup R_H^-$). As an example, the set $Ax_G = \{R7, R8, L7t, L8t\}$ corresponds to the $\left|Tr_G(L_G^-)\right| + \left|R_G^-\right| = 4$ uncovered formulae of Example 1. These 4 formulae are listed at the bottom of the leftmost column of Table **??**, which also shows the candidate blends for the particular generalization G of that example.
   For $H \subseteq G$, the set $R_H^+ \cup Ax_H \in \mathcal{B}_H$ would be the ideal blend that can be built using the (possibly relaxed) generalization H, but it might be inconsistent. So, in this (blending) step we also compute the set MaxCon of maximal consistent blends $B \in \mathcal{B}_H$ such that $R_H^+ \subseteq B \subseteq R_H^+ \cup Ax_H$. For the running example, this involves exploring the 16 theories of the lattice $\mathcal{L}_{\mathcal{B}_G}$ depicted in Figure 7. The user of the algorithm decides now if the produced blends are good enough or the search must continue. In the first case we stop. If not, go to the next step which will need the set MinInc of minimally inconsistent subsets of $R_H^+ \cup Ax_H$ that extends R_H^+.
4. Relaxation Reduce the set of symbols covered by the current generalization by shrinking this generalization (some simple heuristics for this step are given below), and return to step 2.

3.2 Blending Algorithms

Now we discuss how steps 3 and 4 can be implemented (steps 1 and 2 are obtained from HDTP). The discussion also explains how Algorithms 1 and 2 work.

In Algorithm 1, we use a simple procedure ComputeBlends which, besides the sets R_H^+ and Ax_H introduced above, needs a list 'Init' of initial blend candidates (so each element of Init extends R_H^+). Init must have the property that every possible blend based on the current generalization H is either a superset or a subset of one

[2] A simplified version of HDTP is used, where substitutions must preserve the arity of symbols.

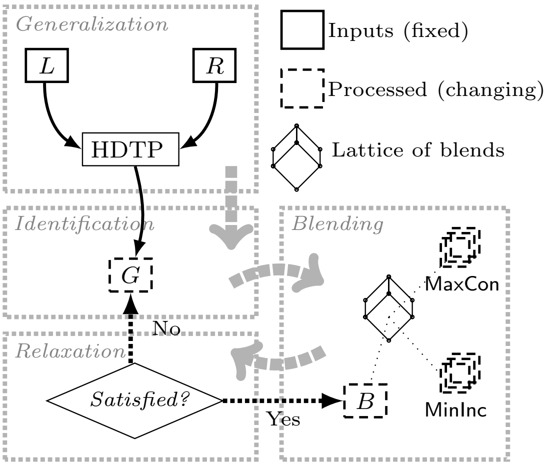

Fig. 6 A depiction of the algorithm's overall logical flow.

Algorithm 1 The ComputeBlends procedure that is used in the blending step.

1: **procedure** ComputeBlends($R_H^+, Ax_H, \mathsf{Init}, direction$)
2: **global** MaxCon $:= \emptyset$
3: **global** MinInc $:= \emptyset$
4: **for each** $T \in \mathsf{Init}$ **do**
5: Explore$\Big(R_H^+, Ax_H, T, direction\Big)$
6: **end for**
7: **end procedure**

of the elements of Init. This —plus the way in which Init will be changed in the relaxation phase (more on this below)— guarantees that the algorithm will find *all* the optimal blends if never asked to stop the search (at the end of step 3). At the very beginning of the process (step 1 above) Init can be initialized, for example, to be the set of theories that extend R_G^+ (a different choice will be used later in our worked example). When a relaxation is needed (step 4 above) a new set Init is computed from MaxCon and MinInc (more on this later). There is a fourth parameter ('*direction*') which is used to direct the search (as explained soon).

The first thing the procedure ComputeBlends does is to initialize as empty two global sets MaxCon and MinInc (lines 2 and 3 in Algorithm 1), which will keep at all times during the search the largest consistent theories and the smallest inconsistent theories, respectively, that have been found up to the moment. After this initialization, the procedure enters into a loop in which for each initial theory T in Init, the procedure Explore (line 5 in Algorithm 1) will populate MaxCon and MinInc. After execution, all blends that contain T or are contained in T, will be "*classified correctly*" by MaxCon and MinInc, i.e. each blend will be subsumed by some theory in MaxCon if it is consistent, or will subsume some theory from MinInc if it is inconsistent (cf. Lemma 1 below). When the loop ends, MaxCon determines precisely the optimal blends.

The Explore procedure is explained in Algorithm 2, which also uses the notations $\uparrow C$ and $\downarrow C$ for a set of theories C. $\uparrow C$ denotes the set of theories that contain some theory from C, whereas $\downarrow C$ denotes the set of theories that are

Algorithm 2 The EXPLORE procedure (cf. Algorithm 1).

1: **procedure** EXPLORE($R_H^+, Ax_H, T, direction$)
2: **if** $T \notin \downarrow\mathsf{MaxCon} \cup \uparrow\mathsf{MinInc}$ **then**
3: **if** T is consistent **then**
4: $\mathsf{MaxCon} := \{T\} \cup \{M \in \mathsf{MaxCon} \mid M \not\subseteq T\}$
5: **else**
6: $\mathsf{MinInc} := \{T\} \cup \{M \in \mathsf{MinInc} \mid T \not\subseteq M\}$
7: **end if**
8: **end if**
9: **if** $T \in \downarrow\mathsf{MaxCon}$ **and** $(direction \in \{up, both\})$ **then**
10: **for each** $Axiom \in (Ax_H \setminus T)$ **do**
11: EXPLORE$\Big(R_H^+, Ax_H, T \cup \{Axiom\}, up\Big)$
12: **end for**
13: **else if** $T \in \uparrow\mathsf{MinInc}$ **and** $(direction \in \{down, both\})$ **then**
14: **for each** $Axiom \in T \setminus R_H^+$ **do**
15: EXPLORE$\Big(R_H^+, Ax_H, T \setminus \{Axiom\}, down\Big)$
16: **end for**
17: **end if**
18: **end procedure**

contained in some theory from C; $\updownarrow C$ is $\uparrow C \cup \downarrow C$. As a first step in EXPLORE (cf., lines 2 to 8 in Algorithm 2), if T is not yet classified by MaxCon or MinInc, consistency of T is checked and either MaxCon or MinInc is updated accordingly. If T is consistent (inconsistent), a recursive upwards (downwards) search towards extensions (subsets) of T is initiated. The upward and downward searches are performed unless the '*direction*' parameter prohibits them. The calls to EXPLORE made when working with the first, strongest generalization G use always the direction '*both*', with the effect that upwards and downwards searches are allowed. In the case of calls to EXPLORE after a 'relaxation' has been made, the direction is set to *up* (the reasons for this will be explained later)[3].

3.3 More Explanations and Results

The above claims about EXPLORE follow from the next result, in which R_H^+ and Ax_H are fixed and the words "theory blend" refer to sets T such that $R_H^+ \subseteq T \subseteq R_H^+ \cup Ax_H$. Also, we will say that *MaxCon and MinInc classify correctly* if all the elements of MaxCon are consistent theory blends and all elements of MinInc are inconsistent theory blends.

Lemma 1. The following pre- and post conditions hold true of the operation of EXPLORE$\big(R_H^+, Ax_H, T, direction\big)$, for all theory blends T:
(1) If all consistency checks can be accomplished, the procedure will terminate.
(2) If MaxCon and MinInc classify correctly before calling EXPLORE, then the same holds afterwards.
(3) If a theory blend B is classified correctly by MaxCon and MinInc before calling EXPLORE, then the same holds after executing EXPLORE.
(4) If $direction = up$ and MaxCon and MinInc classify correctly before calling

[3] There are standard ways to improve the efficiency of the above procedure (using ordered lists, for example), but such discussion would lead us away from the main focus of this paper.

		TR	$T1$	$T2$	$T3$	$T4$	$T5$	$T6$	$T7$	$T8$	$T9$	TL
$x \leq_R x$	$(R1)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$x \leq_R y \wedge y \leq_R z \to x \leq_R z$	$(R2)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$x \leq_R y \vee y \leq_R x$	$(R3)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$0 \leq_R x$	$(R4)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$x +_R y = y +_R x$	$(R5)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$(x +_R y) +_R z = x +_R (y +_R z)$	$(R6)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$x +_R 0 = x$	$(R7)$	⊠	⊠			⊠	⊠	⊠	⊠		⊠	
$x <_R y \to \exists z : (x <_R z \wedge z <_R y)$	$(R8)$	⊠	⊠	⊠	⊠	⊠		⊠		⊠		
$\neg(x +_R 0 \leq_R x)$	$(L7t)$		⊠	⊠	⊠		⊠	⊠	⊠			⊠
$x \leq_R y \wedge y \leq_R x + 0 \to y = x \vee y = x +_R 0$	$(L8t)$				⊠	⊠		⊠	⊠	⊠	⊠	⊠
Consistent:		Y	N	Y^*	N	Y^*	N	N	N	Y	Y	Y^*

Table 1 The table shows some of the theories in the search space of possible blends. Maximal consistent theories are starred. Formulae $L7t$ and $L8t$ result from transferring the uncovered formulae of axiomatization L, according to generalization G.

EXPLORE, then $\uparrow T$ is classified correctly by MaxCon and MinInc after executing EXPLORE.
(5) If $direction = up$ and MaxCon and MinInc classify correctly before calling EXPLORE, then $\downarrow T$ is classified correctly by MaxCon and MinInc after executing EXPLORE.
(6) If $direction = both$ and MaxCon and MinInc classify correctly before calling EXPLORE, then $\updownarrow T$ is classified correctly by MaxCon and MinInc after executing EXPLORE.

Proof. To show (1) notice first that the recursion will only occur with strictly larger ($direction = up$) or strictly smaller ($direction = down$) values for T. As the size of T is limited by R_H^+ and $R_H^+ \cup Ax_H$ the claim follows. (2) follows directly, as MaxCon is only changed when a consistent blend T is added. The case for MinInc is analogous. (3) Let B be a consistent blend. By assumption $B \in \downarrow \mathsf{MaxCon}$ before executing EXPLORE. MaxCon is only changed if T is consistent but $T \notin \mathsf{MaxCon}$, in which case MaxCon will become $\{T\} \cup \{M \in \mathsf{MaxCon} \mid M \not\subseteq T\}$. Now either $B \subseteq T$ or $B \subseteq M \in \mathsf{MaxCon}$ with $M \not\subseteq T$. In both cases B is classified correctly by the new MaxCon. (4) We proceed by induction on the cardinality of $Ax_H \setminus T$. If T is inconsistent, no recursive call to EXPLORE is made. If $T \in \uparrow \mathsf{MinInc}$ there is nothing to prove. If $T \notin \uparrow \mathsf{MinInc}$, observe that T will be added to MinInc, so at the end of the procedure $\uparrow T$ will be classified correctly by MaxCon and MinInc. Now, if T is consistent and $T \notin \downarrow \mathsf{MaxCon}$, then T will be added to MaxCon. Then, for each element A of $Ax_H \setminus T$, a call $\textsc{Explore}\big(R_H^+, Ax_H, T \cup \{A\}, up\big)$ will be made. By inductive hypothesis, after all these calls, every $\uparrow(T \cup \{A\})$ is classified correctly by MaxCon and MinInc, and so (since T is also classified correctly) $\uparrow T$ is classified correctly. (5) The argument is analogous to that for (4), now using induction on the cardinality of $T \setminus R_H^+$. (6) If T is consistent, an argument very close to that of (4) shows that $\uparrow T$ is classified correctly, so $T \subseteq T'$ for some $T' \in \mathsf{MaxCon}$. Then $\downarrow T$ is classified correctly as well. A similar argument applies if T is inconsistent. $\square$

4 "Relaxation" Revisited

As our framework stands, the evaluation of blends in step 3 (i.e., "blending") and the decision to stop or continue with a relaxation, is mandatorily an interactive step that the user decides. After a current generalization $H \subseteq G$ has been dealt with, and if the relaxation step is needed, it is important to find a good weakening K of H and a good set Init with which to continue to step 2 (i.e., "identification"). In principle, the framework allows for an interactive implementation where the user decides which weakened generalization to use next, or for an implementation that uses automated heuristics, such as building a weakened generalizations for which: (i) only one old symbol mapping is dropped, and (ii) the fewest number of axioms become uncovered under the new generalization. In any case, once a weakened generalization $K \subseteq H$ has been fixed, the previously found MaxCon and MinInc sets are used to compute an appropriate new Init set as follows. Let Tr_H and Tr_K be the old and new translation functions. To form the set Init, for each T in MinInc (and optionally for every minimal extension of MaxCon) add to Init the theory that results from replacing in T every formula of the form $Tr_H(\phi)$ in R_H^- by $Tr_K(\phi)$. This new Init is good in that every optimal blend for the weakened generalization will be an extension of one of the Init elements. This is why the exploration, after some relaxation has been made, can be constrained to be upwards only.

4.1 Regarding ComputeBlends and Explore

Let us denote by $\mathbf{S}_H$ the subspace of blend diagrams of the form $\langle H, \lambda_H, \rho_H, B\rangle$, where $H \subseteq G$ and $B \in \mathcal{B}_H$. With $\mathbf{H} = \langle H, \lambda_H, \rho_H\rangle$ being fixed, it is clear that $\mathbf{S}_H$ is isomorphic to the power set of Ax_H. The algorithm ComputeBlends, corresponding to step 3 of our blending procedure, says how to move around a given $\mathbf{S}_H$ so as to find all the blend diagrams $\langle H, \lambda_H, \rho_H, B\rangle$ for which B is maximally consistent. Note that, before the relaxation step is ever reached, our blending procedure consists of exploring $\mathbf{S}_G$, and Lemma 1 shows that ComputeBlends indeed finds all of the optimal blends in this subspace (with the choice of parameters with which Explore is invoked). Notice also that while restricted to stay within a subspace $\mathbf{S}_H$, we cannot change compression, because the generalization is fixed, so optimal blends in $\mathbf{S}_H$ are fully determined by maximal informativeness (i.e., maximal consistency of the last component B). The fact that after using Explore for the first time MaxCon and MinInc contain *all* the maximal consistent B's and *all* the minimal inconsistent B's from $\mathbf{S}_G$, respectively, follows from Lemma 1 together with the condition that upon the beginning of the procedure, Init must be an antichain of the power set of Ax_G. That is, every $B \in \mathcal{B}_G$ from $\mathbf{S}_G$ is a subset or a superset of an element of Init.

To explain what happens when our procedure enters into relaxation stages, it will be convenient to picture the full search space as being composed of all the subspaces $\mathbf{S}_H$, ordered according to H, so that $\mathbf{S}_H \preceq \mathbf{S}_K$ if and only if K is a relaxation of H. Notice that this entails that $K \subseteq H$ and therefore $|Ax_H| \leq |Ax_K|$. Thus, if $\mathbf{S}_H \preceq \mathbf{S}_K$ then $\mathbf{S}_H$ is a smaller space (in terms of cardinality) than $\mathbf{S}_K$, since these spaces are isomorphic as lattices to the power sets of Ax_H and Ax_K, respectively. Now, assume that all the optimal blends within an $\mathbf{S}_H$ space have been found and, even more, all of the maximal consistent $B \in \mathcal{B}_H$ are stored in

MaxCon while all of the minimal inconsistent $B \in \mathcal{B}_H$ are in MinInc. We want to move now to the larger (relaxed) space $\mathbf{S}_K$, where $K \subset H$, and find all the optimal blends in that subspace. One way to do it would be to identify a new Init that is an antichain of the power set of Ax_K and proceed exactly as in the case of exploring the initial $\mathbf{S}_G$, in a mixed up and down direction. We will show, however, that the antichain condition on Init is not really needed anymore, and the results obtained for $\mathbf{S}_H$ allow us to construct a new Init with the property that every optimal blend must be 'above' one of the elements of Init in $\mathbf{S}_K$. Thus, we will only need to focus on some "subregions" of $\mathbf{S}_K$. The strategy follows from Definition 2 and Lemma 2, where we use the notation $[H, B]$ as a shortcut for $\langle H, \lambda_H, \rho_H, B\rangle$; an element of $\mathbf{S}_H$.

Definition 2. Let $\mathbf{K} = \langle K, \lambda_K, \rho_K\rangle$ be a relaxation of $\mathbf{H} = \langle H, \lambda_H, \rho_H\rangle$ and let $[H, B] = \langle H, \lambda_H, \rho_H, B\rangle$ be an element of $\mathbf{H}$. The splitting of B under $\mathbf{K}$ (denoted $split_K[H, B]$) is the set of all $[K, B'] = \langle K, \lambda_K, \rho_K, B'\rangle$ such that:

1. all the elements (formulae) of B that keep being covered by the relaxed generalization $\mathbf{K}$ and all the elements of B that were *not* originally covered by $\mathbf{H}$ belong to B', and
2. any other element of B' must belong to $(R^+_H \setminus R^+_K) \cup Tr_H(L^+_H \setminus L^+_K)$.

So, suppose that upon relaxing from $\mathbf{H}$ to $\mathbf{K}$, there is exactly one formula φ in $B \cap R^+_H$ which stops being covered by K. That is, unlike before the relaxation, φ is not anymore a simple renaming of a formula in L^+_K. Then there will be exactly four elements of $split_K[H, B]$, namely, the theories $(B \setminus \{\varphi\}) \cup C$, where $C \subseteq \{\varphi, \varphi_L\}$ and φ_L is obtained from φ by changing the left signature symbols that are no longer aligned by $\mathbf{K}$ into their corresponding symbols in the left signature (according to $\mathbf{K}$). Similarly, it is easy to see that if the relaxation leads to n formulae in B which are not covered anymore by the new generalization K, then $split_K[H, B]$ will have 2^{2n} elements. It is just an observation that if $\mathbf{K} = \langle K, \lambda_K, \rho_K\rangle$ is a relaxation of $\mathbf{H} = \langle H, \lambda_H, \rho_H\rangle$ and $[K, B] \in \mathbf{S}_K$, then there is a unique element $[H, B'] \in \mathbf{S}_H$ such that $[K, B] \in split_K[H, B]$. We call $[H, B']$ the *contraction* of $[K, B]$ under $\mathbf{H}$.

Lemma 2. Let $\mathbf{K} = \langle K, \lambda_K, \rho_K\rangle$ be a relaxation of $\mathbf{H} = \langle H, \lambda_H, \rho_H\rangle$.

1. If $[K, B'] \in \mathbf{S}_K$ and B' is inconsistent, then B in the contraction $[H, B]$ of $[K, B']$ under $\mathbf{H}$ is also inconsistent.
2. If $[H, B] \in \mathbf{S}_H$, B is consistent, and $[K, B'] \in split_K[H, B]$, then B' is also consistent.

Proof. As for (1), it is a simple observation that, if there exist a way to formally derive a contradiction from B' using first order logic, then there also exist a derivation of a contradiction from B, since B is equivalent to B' with the addition of some equality and/or equivalence axioms of the form

$$\forall \overline{x}(f(\overline{x}) = g(\overline{x})) \text{ or } \forall \overline{x}(R(\overline{x}) \leftrightarrow T(\overline{x}))$$

which capture the alignment (identification) of more symbols by $\mathbf{H}$.

Part (2) follows from the fact that saying that $[K, B'] \in split_K[H, B]$ is the same as saying that $[H, B]$ is the contraction of $[K, B']$ under $\mathbf{H}$. The contrapositive of part (1) tells us that B being consistent entails that B' is consistent as well. $\square$

Back to our blending procedure, Lemma 2 tells us that, if we are after the list of all $[K, B'] \in \mathbf{S}_K$ that are optimal blends (i.e., maximally informative and compressed), it will be enough for us to explore the regions of $\mathbf{S}_K$ that are *above* (i.e., preceed) one of the elements of the set MinInc_H in the informativeness order (see Equation 1).

$$\mathsf{MinInc}_H = \bigcup \{ \mathit{split}_K[H, B] : B \text{ minimally inconsistent in } \mathbf{S}_H \} \tag{1}$$

For if B' is consistent but B in the contraction of $[H, B]$ of $[K, B']$ was also consistent, then $[K, B']$ would not maximally compressed. So the B in the contraction of $[K, B']$ under $\mathbf{H}$ must be inconsistent.

Now, should we want to relax $\mathbf{K}$ after being done with searching optimal blends in $\mathbf{S}_K$, we'd like to have the list of all $[K, B'] \in \mathbf{S}_K$ where B' is minimally inconsistent, to be used in that new relaxation stage. But, when can B' be minimally inconsistent? Again, Lemma 2 gives us that if B' is inconsistent, then B in the contraction of $[K, B']$ under $\mathbf{H}$ must be inconsistent. So, same as the search for optimal blends, the search of minimally inconsistent $B's$ from $\mathbf{S}_K$ can be restricted to the region of the subspace $\mathbf{S}_K$ formed by the blends that are more informative that at least one of the elements of MinInc_H.

The above are the reasons why in all the relaxation stages procedure EXPLORE is called only with direction parameter *up* (there is downwards exploration only in the initial pre-relaxation stage). Since each $\mathit{split}_K[H, B]$ has a minimum element, it would be enough in the relaxation stage to initialize Init with all those minimum elements and explore upwards in $\mathbf{S}_K$.

4.2 Some Considerations of Efficiency

The above remarks show that the pruning we make when fixing a (relaxed) generalization H and exploring the associated subspace $\mathbf{S}_H$ is not only a pruning of $\mathbf{S}_H$ itself, but is simultaneously a pruning of all the relaxed spaces $\mathbf{S}_K$ that might be potentially explored in later stages by relaxing H to $K \subset H$. Remember that the subspaces that result from relaxations are larger in size. In fact, remember that an $\mathbf{S}_H$ is isomorphic to a power set, so the size of these spaces grows exponentially with the number of axioms that need to be "split" when doing a relaxation. So it seems wise to proceed as we currently do, that is, to start by exploring the space $\mathbf{S}_G$ associated with the most compressed generalization (therefore the smaller subspace), knowing that we are pruning much larger spaces at the same time. These same ideas also justify a heuristics for choosing first, among all possible relaxations of a given $\mathbf{H}$, those that would yield the smallest size for the induced *split* sets.

In spite of all this, the complexity, in the worst case, for exploring the initial space $\mathbf{S}_G$ keeps being terrible if the task is really to find *all* the optimal blends, as one can come up with examples of $Ax_G = \{\varphi_1, \ldots \varphi_{2n}\}$ such that each formula of the form $\bigwedge_{\psi \in G} \psi \wedge \bigwedge_{\psi \in C} \varphi$ is consistent for each subset C of Ax_G with $|C| = n$, but $\bigwedge_{\psi \in G} \psi \wedge \bigwedge_{\psi \in D} \varphi$ is inconsistent for each subset D of Ax_G with $|D| = n + 1$. Ulf's example that Maricarmen mentioned is needed here. Such a case would yield a list MinInc with $\dfrac{(2n)!}{(n!)^2}$ elements, a quantity that is asymptotically similar to $\dfrac{4^n}{\sqrt{\pi n}}$. This is an intrinsic problem, which of course motivates the task of trying to find heuristics for exploring the search space in a more directed way that would lead to finding the "most interesting" blends first, so that in many cases one could stop at an interesting finding and not try to complete the search for all the remaining optimal blends in the space. Our algorithm involves testing theories in first-order logic with equality for inconsistency; this is well-known to be undecidable in general. In our examples the inconsistencies will be discovered quickly[4], but in more elaborate situations, a resource-bounded check for inconsistency may model reasonably well the experience of mathematicians who can work productively with theories that are believed to be consistent and later revise their results in case an inconsistency is found. Research on Nelson Oppen methods (see [16] for a survey) reveals conditions under which the satisfiability and decidability of two theories is preserved when taking their union. The basic case requires the signatures of the two theories to be disjoint, but this can sometimes be relaxed. Some of these technical results might end up being useful to our work.

5 Worked Example

To illustrate the algorithm and suggest at least one improvement to it, we come back to take the theories shown in Table **??**. Remember that L is based on the additive natural numbers (starting from 1) and R on the non-negative rational numbers. Thus, the notion of 'number' in L is discrete with least element 1, whereas in R it is dense with least element 0 (as the neutral element for addition). We will find all the optimal blends of L and R. The example shows that our approach isolates just a few optimal blends among many candidates, and that the short list includes (although not exclusively) the ones that one would expect a mathematician to judge as most interesting.

The first stage of the procedure was already partially described in the previous section. It explores the potential blends based on the generalization G of Table **??**. Figure 7 shows a lattice of the blends and Table **??** lists the axioms of each candidate blend. Our set of initial theories will be formed by the minimal extensions of theory R and the minimal extensions of (the transferred version of) theory L. That is, $\mathsf{Init} := \{T1, T3, T7, T4\}$. The sets MaxCon and MinInc are initialized as empty and we start to explore the initial theories. The first is $T1$, which is inconsistent:

$$x +_R 0 = x \qquad (R7)$$
$$\neg(x +_R 0 \leq_R x) \qquad (L7t)$$
$$\neg(x \leq_R x) \qquad \text{(Substitution)}$$
$$x \leq_R x \qquad (R1)$$

The last two lines are clearly contradictory. The algorithm orders to add $T1$ to MinInc. However, knowing that the inconsistency arises from only the axioms $R1, R7$, and $L7t$, it is better to add the smaller $T5$ to MinInc than adding $T1$ itself. Thus, $\mathsf{MinInc} := \{T5\}$.

Now, as the algorithm prescribes, we recursively explore (downwards) every theory obtained from $T1$ by deleting one axiom. These theories are $TR, T2$, and

[4] HDTP and an implementation of the blending phase module are available on request. The blending module uses *prover9* to check for consistency.

$T5$: TR is consistent and $T5 \not\subseteq TR$, so MaxCon := $\{TR\}$; $T2$ is consistent, not contained in TR, and does not extend $T5$, then we update MaxCon := $\{TR, T2\}$; and $T5$ extends the only member of MinInc, so we do nothing. This ends the analysis of $T1$.

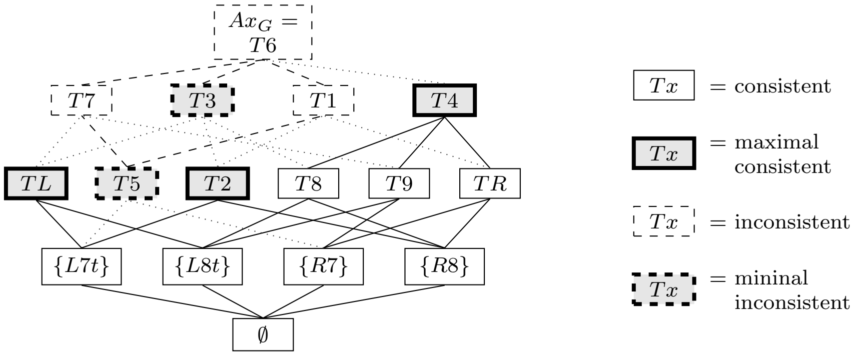

Fig. 7 The lattice $\mathcal{L}_{\mathcal{B}_G}$ of the 'blends' that appear in the given example.

The second initial theory is $T3$. This theory is not a subset of TR or $T2$, and does not extend $T5$. In addition it is inconsistent, as shown by the third and last lines of the following proof, which uses all the axioms of $T3$ not covered by the generalization.

$$\neg(x +_R 0 \leq x) \qquad (L7t)$$
$$\neg(x +_R 0 \leq x) \rightarrow \exists z : (x <_R z \wedge z <_R x +_R 0) \qquad (R8)$$
$$x <_R z \wedge z <_R x +_R 0 \qquad \text{(FOL)}$$
$$\neg(z \leq_R x) \wedge \neg(x + 0 \leq_R z) \qquad (\text{Def. } \leq_R)$$
$$x \leq_R z \wedge z \leq_R x +_R 0 \qquad (\text{FOL} + R3)$$
$$z = x \vee z = x +_R 0 \qquad (\text{MP with } L8t)$$
$$z \leq_R x \vee x +_R 0 \leq_R x \qquad (\text{FOL} + R1 + \text{Def. } \leq_R)$$

We update MinInc:= $\{T5, T3\}$, and recursively explore (downwards!) every theory obtained from $T3$ by erasing one axiom, namely $TL, T2$, and $T8$:

1. TL is consistent and does not extend TR nor $T2$, then MaxCon:= $\{TR, T2, TL\}$. We are in the "downwards" mode, so we stop.
2. $T2$ is a member of MaxCon, so we stop.
3. $T8$ is consistent and not contained in a member of MaxCon. We set MaxCon:= $\{TR, T2, TL, T8\}$. Again, we are in the "downwards" mode, so this branch stops.

This ends the analysis of $T3$, the second initial theory.

The third initial theory is $T7$, but the analysis of it stops immediately as it extends $T5 \in$ MinInc. We are left with the initial theory $T4$, which is consistent and not contained in MaxCon. Then MaxCon is updated by deleting the subsets of $T4$ (TR and $T8$) and adding $T4$: MaxCon := $\{T4, T2, TL\}$. Then we recursively explore (upwards) for possible consistent extensions of $T4$. The only proper extension of $T4$ is $T6$, which extends elements of MinInc. The first stage of the algorithm ends thus:

- **Solutions**: $T2$, $T4$, and TL.
- **Minimally inconsistent theories**: $T5$ and $T3$.

Note that TL is just a signature renaming of theory L, $T4$ a case of analogical transfer but not a *proper* blend, and $T2$ a proper blend intuitively describing the rationals larger than some nonzero number, which is not more interesting than the rationals starting with zero, to which L corresponds. It is then fair to assume that the user will decide to continue the search. In the second search stage, some of the contradictions found in stage 1 will be avoided by weakening the signature of the generalization in the relaxation step. The weakening heuristics described in the previous section suggest dropping the identification between 0 and 1, as this is the dropping that would diminish coverage the least. The new generalized theory changes only in that $(G4)$ is not an axiom of it anymore. The result of transferring all of the axioms of axiomatization L to the R side involves the introduction of a new symbol of constant (1) to the R-side; cf. Table **??**.

		$T30$	$T50$	$T51$	$T52$	$T53$	$T10$	$T11$	$T12$	$T13$	$T62$	$T72$
$(R1)-(R3),(R5),(R6)$		⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$0 \leq_R x$	$(R4)$				⊠	⊠			⊠	⊠	⊠	⊠
$x +_R 0 = x$	$(R7)$		⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$x <_R y \rightarrow \exists z : (x <_R z \wedge z <_R y)$	$(R8)$	⊠					⊠	⊠	⊠	⊠	⊠	
$1 \leq_R x$	$(L4tt)$			⊠		⊠		⊠		⊠		
$\neg(x +_R 1 \leq_R x)$	$(L7tt)$	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠
$x \leq_R y \wedge y \leq_R x + 1 \rightarrow$ $y = x \vee y = x +_R 1$	$(L8tt)$	⊠								⊠	⊠	⊠
Consistent:		N	Y	N	Y	N	Y	N	Y^*	N	N	Y^*

Table 2 Formulae $Lxxx$ result from transferring the uncovered formulae of L according to the weakened generalization that does not identify 0 and 1. Maximal consistent theories are starred.

The set of initial theories will consist of the smallest versions, under the new signature, of the theories associated with the elements of MinInc from stage 1. More in detail, under the new signature there are four versions of each old theory Tj from the first stage. We call them $Tj0$, $Tj1$, $Tj2$, or $Tj3$ depending on which subset of $\{R4, L4tt\}$ they contain: $Tj0$ includes no element from $\{R4, L4tt\}$, $Rj1$ includes only $L4tt$, $Rj2$ includes only $R4$, and $Rj3$ includes the two axioms. Only some of these theories are shown in Table **??**. Our set of initial theories in this stage will then be Init $:= \{T30, T50\}$. The sets MaxCon and MinInc are reset to the empty set.

Every maximally compressed solution blend with respect to the new generalization must extend one of the initial theories. We explore each one of these initial theories in the "upwards" mode. We start with $T30$. This theory is inconsistent because the proof used in stage 1 to see that $T3$ is inconsistent still goes through when using 1 instead of 0 throughout, and $L7tt$ instead of $L7t$. We update MinInc $:= \{T30\}$.

Then we test the second and last initial theory, $T50$. The theory is consistent but may not be maximal. We update MaxCon$:= \{T50\}$, and explore $T50$'s minimal extensions:

1. $T51$ is inconsistent and does not extend $T30$, therefore $\mathsf{MinInc} := \{T30, T51\}$.
2. $T10$ is consistent and extends $T50$. Set $\mathsf{MaxCon}{:=}\ \{T10\}$ and explore the three minimal extensions of $T10$, thus: $T60$ and $T11$ extend the elements $T30$ and $T51$ of MinInc, so nothing is done in these cases; and $T12$ is consistent and properly extends $T10$. Thus, we update $\mathsf{MaxCon}{:=}\ \{T12\}$ and test the minimal extensions of $T12$. There are only two cases of such a minimal extension: Adding $L4tt$ to $T12$ yields a theory that extends the element $T51$ of MinInc; and Adding $L8tt$ yields the theory $T62$, which is inconsistent because it extends $T30 \in \mathsf{MinInc}$.
3. $T70 = T50 \cup \{L8tt\}$ is consistent. So we update $\mathsf{MaxCon}{:=}\ \{T12, T70\}$, and explore the minimal extensions of $T70$. They are: $T60$ (which extends $T30 \in \mathsf{MinInc}$), $T71$ (which extends $T51 \in \mathsf{MinInc}$), and $T72$ (maximal consistent). After these explorations, $\mathsf{MaxCon}{:=}\ \{T12, T72\}$, and $\mathsf{MinInc}{:=}\ \{T30, T51\}$.
4. $T52$ is a subset of $T12 \in \mathsf{MaxCon}$, so we stop.

The second stage ends with new solutions $T12$ and $T72$, which, we claim, are the two mathematically interesting blends of the given theories: there are distinguished numbers 0 and 1, with 0 the unit for addition, and 1 strictly greater than 0; $T72$ is discrete, with a zero element immediately below 1, while $T12$ is dense, with a distinguished unit size.

6 Concluding Discussion

We presented a new algorithmic way of performing theory blending, based on the HDTP framework. Our approach is inspired by Goguen's treatment of CB, but differs from his in various aspects. First, our system generally outputs fewer blends focusing on maximal informativeness and compression as optimality criteria. By this we capture some aspects from [8]'s "optimality principles" for blends. Second, our algorithm uses only the weakenings of a fixed generalization, while Goguen seems to require the exploration of many (possibly mutually incompatible) starting generalizations. Our account also differs from that of [19], as there mappings "do not have to rely on similarity: they can present conflicts that are striking, surprising or even incongruous" [19, p. 90].

Our approach performs CB as theory blending. It therefore is especially appealing for applications in mathematics (such as the automated creation of mathematical concepts and conjectures) and logic-based AI. We demonstrated how traditional optimality criteria for CB can be spelled out in this setting. Also, we can add consistency as a further criterion to judge the quality of blends. As discussed, some relaxations of our algorithms (e.g. using bounded checks) may yield a better fit with human performance. We will also need to study more heuristics for the generalization relaxation stage, since they will affect the order in which optimal blends will be detected, and so the time needed to make the mathematically-oriented user satisfied by the produced blends — the work in [7] is relevant here.

Other algorithmic accounts are given, for instance, in [19], where the CB mechanism uses a parallel search engine based on genetic algorithms, or in [12], sketching the blending of logical theories within a distributed ontology setup. Further work on CB is contained in [14] where the authors present a rule-based system for counterfactual reasoning in natural language. These examples are mostly addressing problems from linguistics or philosophy.

A related approach to search for blends in a similar framework is described in [7]. The authors use Answer Set Programming to compute a generic space for given input spaces; tools from the Hets system [18] are used to compute blends, and check for consistency of the resultant theories. The subsequent weakening of input theories is guided by ASP, until consistent blends are found. The authors work with input spaces with *prioritised* content (priorities for predicates, axioms and so on), indicating the relative importance of aspects of the given theories. This is an important aspect of mathematical blending, as some conceptual aspects are regarded as more important to a given mathematical concept than others. There are evaluation metrics provided; these are intended to measure the quality of the resultant blend. The priorities guide a heuristic search for good blends. Unlike in our approach, there is no attempt to take into account all consistent blends.

Ideally, we would like to work with such prioritised inputs, and also make use of the global properties of the MinInc, MaxCon so as to reduce search; we will investigate this possibility in the future.

Our interest in this topic lies in particular in the blending of mathematical theories, as a means of understanding certain developments in the history of mathematics, as described by Alexander [2], and also as part of general mathematical cognition, as suggested by Lakatos [13].

References

1. Abdel-Fattah, A.M.H., Krumnack, U., Kühnberger, K.U.: The Importance of Two Cognitive Mechanisms in Analyzing Counterfactuals: An Implementation-Oriented Explication. In: Advances in Cognitive Systems, Baltimore, Maryland, USA (2013)
2. Alexander, J.: Blending in Mathematics. Semiotica **2011**(187), 1–48 (2011)
3. Besold, T.R., Kühnberger, K.U., Plaza, E.: Analogy, Amalgams, and Concept Blending. In: Proceedings of the Third Annual Conference on Advances in Cognitive Systems (ACS 2015), Poster Collection. CogSys.org (2015)
4. Besold, T.R., Plaza, E.: Generalize and Blend: Concept Blending Based on Generalization, Analogy, and Amalgams. In: Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015). Brigham Young University (2015)
5. Confalonieri, R., Corneli, J., Pease, A., E., P., Schorlemmer, M.: Using argumentation to evaluate concept blends in combinatorial creativity. In: Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015). Brigham Young University (2015)
6. Coulson, S.: Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction. Cambridge University Press, Cambridge (2006)
7. Eppe, M., Maclean, E., Confalonieri, R., Schorlemmer, M., Kutz, O., Plaza, E.: ASP, amalgamation and the conceptual blending workflow. In: F. Calimeri, G. Ianni, M. Truszczynski (eds.) Logic Programming and Nonmonotonic Reasoning: 13th International Conference, LPNMR 2015, *Lecture Notes in Artificial Intelligence*, vol. 9345. Springer (2015)
8. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books, New York (2002)
9. Gentner, D.: Structure-mapping: a theoretical framework for analogy. Cognitive Science **7**(2), 155–170 (1983)
10. Goguen, J.: Mathematical models of cognitive space and time. In: D. Andler, Y. Ogawa, M. Okada, S. Watanabe (eds.) Reasoning and Cognition: Proc. of the Interdisciplinary Conference on Reasoning and Cognition, pp. 125–128. Keio University Press (2006)
11. Kutz, O., , Neuhaus, F., Mossakowski, T., Codescu, M.: Blending in the hub: Towards a computational concept invention platform. In: Proceedings of the Fifth International Conference on Computational Creativity (ICCC 2015). Jozef Stefan Institute (2014)
12. Kutz, O., Bateman, J., Neuhaus, F., Mossakowski, T., Bhatt, M.: E pluribus unum: Formalisation, Use-Cases, and Computational Support for Conceptual Blending. In: Computational Creativity Research: Towards Creative Machines, *Atlantis Thinking Machines*, vol. 7. Springer (forthcoming in 2015)

13. Lakoff, G., Núñez, R.: Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being. Basic Books, New York (2000)
14. Lee, M., Barnden, J.: A Computational Approach to Conceptual Blending within Counterfactuals. Cognitive Science Research Papers CSRP-01-10, School of Computer Science, University of Birmingham (2001)
15. Li, B., Zook, A., Davis, N., O' Riedl, M.: Goal-driven conceptual blending: A computational approach for creativity. In: Proceedings of the Third International Conference on Computational Creativity (ICCC 2015). University College Dublin (2012)
16. Manna, Z., Zarba, C.: Combining Decision Procedures. In: Formal Methods at the Crossroads: from Panacea to Foundational Support, LNCS 2787, pp. 381–422. Springer (2003)
17. Martins, P., Urbanci, T., Pollak, S., Lavrac, N., , Cardoso, A.: The good, the bad, and the aha! blends. In: Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015). Brigham Young University (2015)
18. Mossakowski, T., Maeder, C., Lüttich, K.: The Heterogeneous Tool Set. In: O. Grumberg, M. Huth (eds.) TACAS 2007, *Lecture Notes in Computer Science*, vol. 4424, pp. 519–522. Springer-Verlag Heidelberg (2007)
19. Pereira, F.C.: Creativity and AI: A Conceptual Blending Approach. Applications of Cognitive Linguistics (ACL). Mouton de Gruyter, Berlin (2007)
20. Plotkin, G.D.: A Note on Inductive Generalization. Machine Intelligence **5**, 153–163 (1970)
21. Schiralli, M., Sinclair, N.: A Constructive Response to "Where Mathematics Comes From". Educational Studies in Mathematics **52**, 79–91 (2003)
22. Schmidt, M., Krumnack, U., Gust, H., Kühnberger, K.U.: Heuristic-Driven Theory Projection: An Overview. In: Computational Approaches to Analogical Reasoning: Current Trends, *Studies in Computational Intelligence*, vol. 548, pp. 163–194. Springer (2014)
23. Veale, T.: Computability as a test on linguistic theories. In: G. Kristiansen, M. Achard, R. Dirven, F. Ruiz de Mendoza Ibanez (eds.) Cognitive Linguistics: Current Applications and Future Perspectives. Mouton DeGruyter, The Hague (2006)
24. Veale, T.: A robust computational model of conceptual blending. In: Proceedings of the Third International Conference on Computational Creativity (ICCC 2015). University College Dublin (2012)
25. Veale, T., O'Donoghue, D.: Computation and blending. Computational Linguistics **11**(3–4), 253–282 (2000). special issue on Conceptual Blending
26. Xiao, P., Linkola, S.: Vismantic: Meaning-making with images. In: Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015). Brigham Young University (2015)