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

Algorithmic Aspects of Theory Blending

**M. Martinez · A. M. H. Abdel-Fattah ·
U. Krumnack · D. Gómez-Ramírez ·
A. Smaill · T. R. Besold · A. Pease ·
M. Schmidt · M. Guhe · K.-U. Kühnberger**

Received: July 31, 2015 / Accepted: March 22, 2016

Abstract In Cognitive Science, conceptual blending has been proposed as an important cognitive mechanism that facilitates the creation of new concepts and ideas by

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

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

Ulf Krumnack
Institute of Cognitive Science, University of Osnabrück, Osnabrück
E-mail: krumnack@uos.de

Danny Arlen de Jesus Gómez-Ramírez
Institute of Cognitive Science, University of Osnabrück, Osnabrück
E-mail: dagomezramir@uni-osnabrueck.de

Alan Smaill
School of Informatics, University of Edinburgh, Edinburgh
E-mail: a.smaill@ed.ac.uk

Tarek Richard Besold
The KRDB Research Centre, Free University of Bozen-Bolzano, Bolzano
E-mail: TarekRichard.Besold@unibz.it

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

Martin Schmidt
Institute of Cognitive Science, University of Osnabrück, Osnabrück
E-mail: martisch@uos.de

Marcus Guhe
School of Informatics, University of Edinburgh, Edinburgh
E-mail: m.guhe@ed.ac.uk

Kai-Uwe Kühnberger
Institute of Cognitive Science, University of Osnabrück, Osnabrück
E-mail: kkuehnbe@uos.de

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 an interactive algorithm for blending within the framework, and provides several illustrating worked examples from mathematics.

Keywords Concept Blending · Heuristic-Driven Theory Projection

1 Introduction

Conceptual blending theory (CB) [12] 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: the latter is comprised of structure from each of the two domains as well as unique structure of its own [9]. The theory has gained popularity as a way of explaining high-level cognitive and linguistic phenomena, such as metaphor [17], analogy [3], and counterfactual reasoning [1], [18]. 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 approach of CB in [12], however, lacks a formal and algorithmic account.

CB is also considered to play a crucial role in mathematical invention and theory development. Lakoff and Núñez [17] present a blending-based account of the origin and development of mathematical ideas, in which human mathematics is grounded in the bodily experience of physical interactions in the world and inheritance or transfer processes 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 [17] has been criticized and further developed by other researchers over the last 15 years (see, e.g., [27] for a reply and further development of the ideas from [17]), 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 are commonly understood as finitely axiomatized theories in a logical language, 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[1] is structured as follows. In the remainder of this introduction we 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 Section 2, we introduce the formal framework that we use to model blending processes, and establish a notion of optimality for blends that is inspired by cognitive criteria. In Sections 3 and 4, we present an algorithm that, given two input theories, searches for all the optimal theory blends. We also prove the correctness and completeness of our search algorithm, and make some considerations of efficiency. As a proof of concept, we illustrate our algorithm with three worked examples in Section 5. Section 6 contrasts and discusses our method with related approaches and Section 7 finally presents our concluding remarks and our plans for future research.

1.1 Computational Accounts of Concept Blending

The earliest computational models of concept blending, [31] and [25], were based on Gentner's structure-mapping theory (SMT) of analogy [13]. 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 Heuristic-Driven Theory Projection [28]), and amalgamation (combining solutions from multiple cases in case-based reasoning), 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. [23] investigated criteria for creative concept blends, by asking participants to rate human-generated concept blends in terms of some of the optimality principles proposed in [12] and other principles connected to creativity. Confalonieri et al. provide an alternative take on the problem [8], 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. [19] 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 it arises. 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

[1] The current paper is an extended and substantially enhanced version of [22].

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

Fig. 1 Goguen's version of concept blending (cf. [14]).

criticism that the input conceptual spaces consist of handcrafted knowledge: in [30], 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, [29], 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 [32] 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 an image which blends the two concepts. The question of what sort of spaces can be blended is considered by Kutz et al. [16], 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.

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 [14]. We base our formal model, elaborated below, on Goguen's logic-based approach. 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 potentially language changing (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 [14] and linked to earlier work on computational aspects of blending in cognitive linguistics (see, e.g., [31]), 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. [12]). 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 far 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 [14], 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 a framework for computational analogy making between many-sorted *first-order* theories, in order to obtain the generalization space 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 will become 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. Our approach establishes theory morphisms between the input theories and the blend theory and shares similarity with specification morphisms and theory morphisms used in the area of (algebraic) software specification. In this field, findings range from complexity theoretic results concerning the existence of signature morphisms as presented in [6] to expansions of conservative extensions of logical theories to specification morphisms as described in [20] and semantic issues concerning theory morphisms mapping theories belonging to different institutions [10]. In contrast to this work, our approach uses theory morphisms as a formal tool for modeling a cognitively inspired application, namely conceptual blending, in order to give a formal approach for the creative invention of new concepts in mathematics, a goal which is significantly different from the field of specification morphisms.

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 [28]. 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.

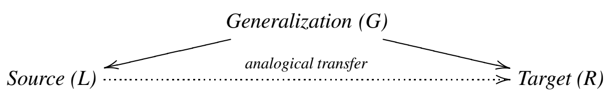

Fig. 2 HDTP's overall approach to creating analogies (cf. [28]).

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 [26] 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, relation and function symbols can also be included in the domain of substitutions (see [28] 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. In the language of theories and theory morphisms, what HDTP does can be described as follows: based on (axiomatizations of) the two input theories, a pair of (derived) signature morphisms $\Sigma_L \xleftarrow{\sigma_L} \Sigma_G \xrightarrow{\sigma_R} \Sigma_R$ is computed, that induces a

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

Axiomatization L		**Axiomatization** R		**Generalization** G	
$x \leq_L x$	$(L1)$	$x \leq_R x$	$(R1)$	$x \leq x$	$(G1)$
$x \leq_L y \wedge y \leq_L z \rightarrow x \leq_L z$	$(L2)$	$x \leq_R y \wedge y \leq_R z \rightarrow x \leq_R z$	$(R2)$	$x \leq y \wedge y \leq z \rightarrow x \leq z$	$(G2)$
$x \leq_L y \vee y \leq_L x$	$(L3)$	$x \leq_R y \vee y \leq_R x$	$(R3)$	$x \leq y \vee y \leq x$	$(G3)$
$1 \leq_L x$	$(L4)$	$0 \leq_R x$	$(R4)$	$a \leq x$	$(G4)$
$x +_L y = y +_L x$	$(L5)$	$x +_R y = y +_R x$	$(R5)$	$x + y = y + x$	$(G5)$
$(x +_L y) +_L z = x +_L (y +_L z)$	$(L6)$	$(x +_R y) +_R z = x +_R (y +_R z)$	$(R6)$	$(x + y) + z = x + (y + z)$	$(G6)$
$\neg(x +_L 1 \leq_L x)$	$(L7)$	$x +_R 0 = x$	$(R7)$		
$x \leq_L y \wedge y \leq_L x +_L 1 \rightarrow y = x \vee y = x +_L 1$	$(L8)$	$x <_R y \rightarrow \exists z : x <_R z \wedge z <_R y$	$(R8)$		

mapping between (combinations of) symbols of the signatures Σ_L and Σ_R. Furthermore, (axioms for) a generalized theory G are proposed in a way that assures that $\sigma_L : G \rightarrow L$ and $\sigma_R : G \rightarrow R$ are theory morphisms.

It should be noted, that all processes in HDTP are syntax-based, and in its most basic form, the axioms of L and R have to be chosen in a way that they exhibit a parallel structure allowing for simple matching. This is a rather strong assumption that can be found in most analogy frameworks, but which seems artificial in many practical applications. In the context of HDTP, a "re-representation" mechanism has been proposed 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. [28, pp. 258]). Thus, syntactically different but semantically equivalent axiomatizations may result in a good generalization. However, as this paper focuses on the blending step, we will not further consider re-representation here, i.e. we expect the axioms of the theories L and R to be given in a suitable form.

We will say that a formula of the input theories is *covered* by G if it is in the image of the projection of G; 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.

Example 1 We will use a first working example based on the theories L and R from Table 1, 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 generalized theory G over the signature $\{a, \leq, +\}$, which reflects the fact that axiom (Li) is structurally like (Ri) when $1 \leq i \leq 6$. In standard mathematical terminology, theory G corresponds to an axiomatization of an *ordered commutative semigroup* with minimal element a. 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 3) occurs when the input spaces are isomorphic, meaning that R is obtained from L via a renaming of symbols of the signature of L to symbols of the signature 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 case (right side of Figure 3) 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. 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.

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

Fig. 3 The two extreme cases of input spaces, along with their generalizations and blends.

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:

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

Some remarks concerning the principles CP and IP are necessary here. Both principles are inspired by Fauconnier and Turner's work: For example, compression in the blend space plays an important role in [12], e.g. in form of how to create compressions, how to compress single relations by scaling, how to compress relations into other relations etc. Due to the fact that Fauconnier and Turner's approach is informal and does not contain any technical details, our principles can be seen as a possible formal manifestation of the underlying ideas in [12]. Furthermore, different degenerated examples of (non-)compressed and (non-)informative blends can be explained using Fig. 3. Whereas a blend $L \oplus R$ is maximally informative, because a maximal subset of the merged formulae is included (although the blend space might be inconsistent) it is nevertheless non-compressed. $L \cong R$ is maximally compressed and maximally informative, because all theories are essentially isomorphic. If we take for the blend in this case not L or R but a proper subset of L (or alternatively R), in the extreme case the empty set, such a blend would be non-informative and non-compressed. Note finally 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 1 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 $\rho_H \subseteq \rho_G$.

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

Fig. 4 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 4 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 R, R^+_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 every 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.

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 right input space when forming blends. However, as will become clear later, the core of our algorithmic approach is unchanged if a different symbol collapsing 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 4. 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:

(i) $\mathbf{H} = \langle H, \lambda_H, \rho_H \rangle$ 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 $H \subseteq G$, $\lambda_H \subseteq \lambda_G$, $\rho_H \subseteq \rho_G$.
(ii) $R_H^+ \subseteq B \subseteq R_H^+ \cup Ax_H$.

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

Now we are ready to present and discuss our overall search strategy, depicted in Figure 5 and further explained in the rest of this section. Given two inputs L and R over first-order signatures Σ_L and Σ_R, respectively, we propose the following 4-stage strategy to find optimal blends.

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 1 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 2, 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 6.
   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.

In search of maximally informative blends, the main idea of this 4-stage general strategy is to scan the search space in "layers". Each layer is determined by a generalization, starting with the fixed generalization initially given by HDTP. Then, each time the relaxation step is encountered, consequent generalizations are relaxed, meaning that the scan starts all over with a weakened relaxation (i.e., those generalizations partially losing their "compression"). This process of "layer scanning" corresponds to the Identification–Blending–Relaxation cycle of steps depicted visually in Figure 5.

---

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

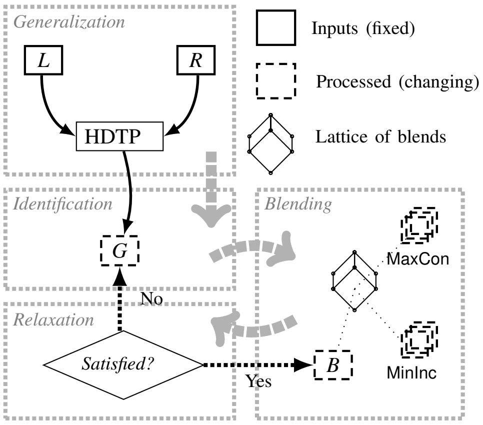

Fig. 5 A depiction of the algorithm's overall logical flow.

3.1 Blending-Stage Algorithms

In the rest of this section, we focus on step 3 only: step 1 is obtained from HDTP, step 2 does not require further explanation, and step 4 will be discussed in Section 4. The pseudocode of step 3 comprises the procedures shown in Algorithms 1 and 2.[3]

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

procedure COMPUTEBLENDS(R_H^+, Ax_H, Init, direction)
  [MaxCon,MinInc] := [$\emptyset$,$\emptyset$]
  for each $T \in$ Init **do**
    [MaxCon,MinInc] = EXPLORE(R_H^+, Ax_H, T, direction, [MaxCon, MinInc])
  end for
  return [MaxCon,MinInc]
end procedure

In Algorithm 1, we have 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 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).

[3] A Prolog implementation of the algorithm is available at `http://www.coinvent-project.eu/en/publicationsmedia/other_media_public_appearances.html`.

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.

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

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

The EXPLORE procedure is given in Algorithm 2, where the notations $\uparrow C$ and $\downarrow C$ are used for a given theory C: $\uparrow C$ denotes the set of theories that contain some theory from C, whereas $\downarrow C$ denotes the set of theories that are 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)[4].

[4] 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.

3.2 Correctness and Completeness of the Blending Stage

The above claims about EXPLORE follow from the next result, in which R_H^+ and Ax_H are fixed and "theory blend" refers to sets T such that $R_H^+ \subseteq T \subseteq R_H^+ \cup Ax_H$.

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* EXPLORE*, then $\uparrow T$ is classified correctly by* MaxCon *and* MinInc *after executing* EXPLORE.
(5) If $direction = down$ 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 $B \in \downarrow$MaxCon before executing EXPLORE. MaxCon is only changed if T is consistent but $T \notin$ MaxCon, in which case MaxCon will become $\{T\} \cup \{M \in \text{MaxCon} \mid M \not\subseteq T\}$. Now either $B \subseteq T$ or $B \subseteq M \in$ MaxCon with $M \not\subseteq T$. In both cases B is classified correctly by the new MaxCon. A similar argument holds for MinInc.

(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 $T \in \uparrow$MinInc there is nothing to prove. If $T \notin \uparrow$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 $T \notin \downarrow$MaxCon, then T will be added to MaxCon. Then, for each element A of $Ax_H \setminus T$, a call 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 $T' \in$ MaxCon. Then $\downarrow T$ is classified correctly as well. A similar argument applies if T is inconsistent. $\square$

Table 2 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*.

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

4 "Relaxation" Revisited

In this section, we study the relaxation stage of our approach. 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 generalization 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. As we will see in this section, 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:

(i) 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
(ii) 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 $\boldsymbol{K} = \langle K, \lambda_K, \rho_K \rangle$ be a relaxation of $\boldsymbol{H} = \langle H, \lambda_H, \rho_H \rangle$.*

1. *If $[K,B'] \in \boldsymbol{S}_K$ and B' is inconsistent, then B in the contraction $[H,B]$ of $[K,B']$ under $\boldsymbol{H}$ is also inconsistent.*
2. *If $[H,B] \in \boldsymbol{S}_H$, 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 exists a way to formally derive a contradiction from B' using first-order logic, then there also exists 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. □

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).

$$\mathsf{MinInc}_H = \bigcup \{split_K[H,B] : B \text{ minimally inconsistent in } \mathbf{S}_H\} \quad (1)$$

For if B' is consistent but B in the contraction of $[H,B]$ of $[K,B']$ was also consistent, then $[K,B']$ would not be maximally compressed. So, 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 would 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 than 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 $split_K[H,B]$ 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 **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 very high if the task is really to find *all* the optimal blends, as one can come up with examples of $Ax_G = \{\varphi_1, \dots \varphi_{2n}\}$ 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 $|D| = n+1$. Such a case would yield a list MinInc with $\frac{(2n)!}{(n!)^2}$ elements, a quantity that is asymptotically similar to $\frac{4^n}{\sqrt{\pi n}}$. 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[5], 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 [21] 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 Examples

5.1 Axiomatizing Arithmetical Properties

To illustrate the algorithm and suggest at least one improvement to it, we come back to take the theories shown in Table 1. Remember that L is based on the additive nat-

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

ural 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 1. Figure 6 shows a lattice of the blends and Table 1 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 adds $T1$ 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, 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 $T5$: 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 $T1$.

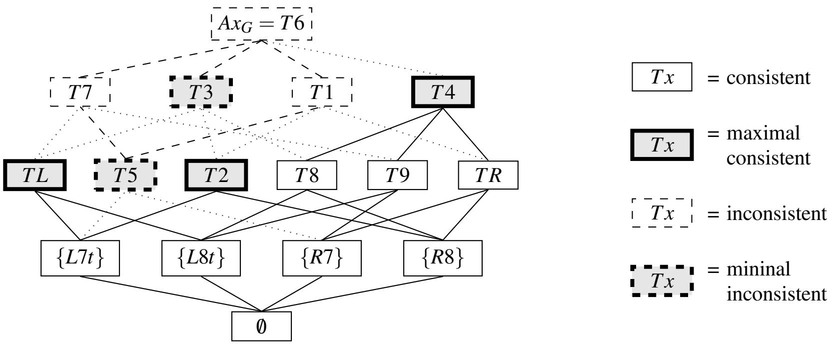

Fig. 6 The lattice $\mathcal{L}_{\mathcal{B}_G}$ of the 'blends' that appear in the given example.

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.

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

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 3.

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

Table 3 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.

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

$\{R4, L4tt\}$ 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 3. Our set of initial theories in this stage will then be $\mathsf{Init} := \{T30, T50\}$. 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 $\mathsf{MinInc} := \{T30\}$.

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

Axiomatization L	**Axiomatization** R
$(L1)$ $(\forall a,b)a *_L b = b *_L a$	$(R1)$ $(\forall a,b)a +_R b = b +_R a$
$(L2)$ $(\forall a,b,c)(a *_L b) *_L c = a *_L (b *_L c)$	$(R2)$ $(\forall a,b,c)(a +_R b) +_R c = a +_R (b +_R c))$
$(L3)$ $(\forall a)a *_L 1 = 1 *_L a = a$	$(R3)$ $(\forall a)a +_R 0 = 0 +_r a = a$
$(L4)$ $(\forall a)1\|_L a$	$(R4)$ $(\forall a)(\exists -a)a + (-a) = (-a) + a = 0$
$(L5)$ $(\forall a,b,c)a\|_L b \to a\|b *_L c$	$(R5)$ $\forall a \neg(0\|_R a)$
$(L6)$ $(\forall a,b,c)a\|_L b \wedge b\|_L c \to a_L c$	$(R6)$ $(\forall a,b,c)(a\|_R b \wedge b\|_R c \to a\|_R c)$
	$(R7)$ $(\forall a)(a \neq 0) \to a\|_R a \wedge a\|_R - a$
	$(R8)$ $(\forall a,b,c)a\|_R b \wedge a\|_R c \to a\|_R (b +_R c)$
	$(R9)$ $(\forall a,b,c)(a *_R b) *_R c = a *_R (b *_R c)$
	$(R10)$ $(\forall a,b,c)a *_R (b +_R c) = a *_R b +_R a *_R c$

Table 4 Explicit axiomatizations of a monoid (and a ring) with (additive) divisibility relations, respectively.

5.2 Commutative Ring with Unity and Compatible Divisibility Relation

We will combine two concepts emerging as a formal union of typical concepts in abstract algebra and number theory. The first concept L is a *commutative monoid with divisibility relation* and the second one R is a *ring with (additive) divisibility relation* (compare Table 4).

We will obtain inconsistent theories in the case that all axioms of L and R are mapped to the blend space and HDTP works with the analogical substitutions

1. $*_L \to +_R; |_L \to |_R$.
2. $*_L \to *_R; |_L \to |_R$.

The first case is obtained when HDTP finds four direct analogical matches between (Lj) and (Rj), for $j = 1,2,3,6$. It is straightforward to prove that in this case $1 = 0$, since 1 would be also a neutral element for the addition operation in R. So, the axioms (R4) and (L5) would generate a contradiction. Besides, the generic space consists of four axioms and, therefore, our algorithm suggests as a maximal consistent blend the concept defined by the axiomatization R plus the axiom $Tr_R(L5)$. Furthermore, the axioms $(R8),(R10)$ and $Tr_R(L5)$ imply that

$$(\forall a,b)(a \neq 0) \to (a|_R b).$$

So, a trivial model for this concept is a ring with two elements 0 and 1. In particular, in this case the relation $|_R$ becomes almost trivial. Effectively, if we consider any ring with at least two elements, the relation $|_R$ defined by $a|_R b$ if and only if $a \neq 0$, is a model of this theory.

The second case appears when HDTP finds as analogical matches $(L2) - (R9)$ and $(L6) - (R8)$. It gives the most representative and rich blend. In fact, the algorithm finds as maximal consistent theory the concept of a *commutative ring with unity and compatible divisibility relation*, which can be seen as a formally interesting combination of the former two concepts, since it could be described as a fundamental notion in commutative algebra, i.e., a commutative ring with unity enriched with a natural arithmetical structure given by a divisibility relation and congruent with the corresponding binary operations.

Axiomatization L	**Axiomatization** R
$(L1)$ $(\forall a,b) a +_L b = b +_L a$	$(R1)$ $(\forall a,b) a +_R b = b +_R a$
$(L2)$ $(\forall a,b,c)(a +_L b) +_L c = a +_L (b +_L c)$	$(R2)$ $(\forall a,b,c)(a +_R b) +_R c = a +_R (b +_R c)$
$(L3)$ $(\forall a) a +_L 0_L = 0_L + a = a$	$(R3)$ $(\forall a) a +_R 0_R = 0_R +_R a = a$
$(L4)$ $(\forall a,b) s(a) = s(b) \rightarrow a = b$	$(R4)$ $(\forall a,b) inv(a) = inv(b) \rightarrow a = b$
$(L5)$ $(\forall a) \neg(s(a) = 0_L)$	$(R5)$ $(\forall a) inv(a) +_R a = 0_R$
$(L6)$ $(\forall a,b) s(a +_L b) = s(a) +_L b$	

Table 5 Axiomatizations for the concepts of quasi-natural numbers as a commutative monoid with successor function (L) and Abelian group with inverse function (R).

5.3 Partial Axiomatization of the Integers

Let us consider as our first concept a partial axiomatization of the natural numbers by means of the addition operation and a successor function (see [5] for a similar axiomatization). Besides, we define as second space the concept of an Abelian group with the axioms explicitly defined through an inverse unary function (see Table 5).

In this case, HDTP finds natural analogical matches between the first four axioms of both theories, which defines the generic space. It generates the signature morphism

$$+_L \rightarrow +_R; \quad s \rightarrow inv; \quad 0_L \rightarrow 0_R.$$

Our algorithm finds the blend consisting of the union of both theories as a minimal inconsistent theory. In fact, $(R5)$ implies $inv(0_R) = 0_R$, which contradicts

$$Tr_G(L5) : (\forall a) \neg(inv(a) = 0_R).$$

Furthermore, the set of maximal consistent theories consists basically of a space isomorphic to L and the space obtained after subtracting the former axiom, which gives the theory of Abelian groups with elements of order at most two. Effectively, the axioms $Tr_R(L6)$ and $(R5)$ imply

$$(\forall a) inv(a +_R a) = inv(a) +_R a = 0_R.$$

So, for any element $a = inv(inv(a))$, it holds $2a = 2inv(inv(a)) = 0_R$.

In conclusion, the resulting blending gives a genuine theory with at least one new algebraic property which cannot be derived from any of the input spaces alone, namely the fact that two is an upper bound for the order of each element of the space.

Now, in the interactive approach as well as in the one using automated heuristics, the next most suitable relaxation to be considered is the following one:

$$+_L \rightarrow +_R; \quad 0_L \rightarrow 0_R.$$

The reason is that it drops the axiom concerning the unary function symbols. So, we obtain in this case just the three first axioms of both concepts in the generic space. Again, the blended theory consisting of all the axioms inherits an equivalent kind of contradiction as in the former substitution.

Now, in the collection of maximal consistent theories our algorithm generates as the most relevant spaces the one obtained by subtracting the same axiom $Tr_R(L6)$, which turns out to be a partial axiomatization of the integers with a successor function s and an implicitly defined predecessor function, namely $inv(s(inv(-))$ (for a similar approach see [5]), which has also an intrinsic mathematical value.

Furthermore, the other two maximal consistent theories are not of the same relevance from a mathematical point of view, since both of them exclude the possibility of comparing simultaneously the common addition operation and the functions s and inv. Therefore, the corresponding models are trivial formal meta-intersections of models coming from sub-theories of each of the input spaces.

6 Evaluation

Blending is widely discussed as an important cognitive mechanism in cognitive linguistics and other branches of cognitive science as it allows to convincingly explain phenomena of learning and creativity. However, most current approaches to blending are on a purely conceptual level, with only few of them implemented, of which most use idiosyncratic formalisms for knowledge representation, tailored to their respective blending mechanisms. For a quantitative evaluation (focusing on the quality of outputs, runtime behavior etc.) we would need comparable other systems. These systems should perform the same sort of blending of mathematical theories with the same type of (in)consistency checking we deal with. To the best of our knowledge there are no systems that are applicable precisely to the same type of problems considered in this paper. Rather such system address related problems, which we briefly describe in the following paragraphs.

The approach presented in this paper is, together with [5], one of the first attempts to perform blending of theories provided in classical first-order logic, addressing problems like inconsistent blend candidates and the relaxation of unsatisfying blends. In comparison to [5], our algorithm has the advantage of being more sensible to consistency and being able to generate semi-automatically the generic space. The model in [5] requires that the user specifies manually the generic space and, after that, it computes using HETS [24] just one possible blend theory in terms of a colimit construction (which has the additional benefit of allowing blended, and therefore, new axioms). Moreover, in the case that the resulting colimit gives an inconsistent theory, the user should modify manually the whole diagram of theories by hand, which is, in general, not the case in our algorithm.

A related approach to searching for blends in a similar framework is described in [11]. The authors use Answer Set Programming (ASP) to compute a generic space for given input spaces. Tools from the HETS system 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 *prioritized* content (priorities for predicates, axioms etc.), 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. 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 prioritized 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.

Further approaches towards creativity and theory formation in mathematics are Colton's HR system [7] and Pereira's blending approach for modeling creativity [25]. The HR system, in its newest version HR3, does not use blending as a methodology, but a mixture of production rules, inductive as well as deductive reasoning mechanisms, and randomization features. Due to the different methodologies the approaches are very hard to compare. Our account also differs from that given in [25], as the mappings in the latter "do not have to rely on similarity: they can present conflicts that are striking, surprising or even incongruous" [25, p. 90].

A final note concerns the choice of the representation language of our approach. The use of first-order logic is motivated by its ability to express many non-trivial mathematical theories. The approach can support mathematicians by suggesting indirectly new concepts with interesting mathematical value (e.g. [15]). Of course, first-order logic also introduces problems, like its semi-decidability which can in principle make the algorithm fail. However, modern theorem provers are often not even restricted to first-order theories but support higher-order logic as well. Furthermore, the overall approach presented in this paper is not dependent on the choice of first-order logic. It can be applied to logical formalisms like Description Logics, which often are decidable and guarantee acceptable runtime at the price of lower expressivity.

7 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 [12]'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 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 [17]. Our approach performs CB as theory blending. Therefore, it 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.

It is desirable to thoroughly evaluate interactive tools supporting working mathematicians in quantitative experiments. Unfortunately, this is currently very hard, because there are not many systems available that can generate new and interesting mathematical concepts (compare Section 6 for an overview), there is no generally accepted set of benchmark problems that can be used for a quantitative evaluation, and the focus and underlying methodology of potential systems coming into consideration varies quite significantly. Nevertheless, we believe that future research should make an attempt to design quantitative experiments in this field. Our own experiments could be used as a first benchmark.

Acknowledgements Some of 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

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. Bou, F., Schorlemmer, M., Corneli, J., Gómez-Ramírez, D., Maclean, E., Smaill, A., Pease, A.: The Role of Blending in Mathematical Invention. In: Proceedings of the 6th International Conference on Computational Creativity (ICCC), *Lecture Notes in Computer Science*, vol. 1342 (2015)
6. Cicerone, S., Parisi-Presicce, F.: On the Complexity of Specification Morphisms. Theoretical Computer Science **189**(1–2), 239–248 (1997)
7. Colton, S., Ramezani, R., Llano, M.T.: The HR3 Discovery System: Design Decisions and Implementation Details. In: R. Kibble (ed.) Proceedings of the 50th Anniversary Convention of the AISB (2014)
8. 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)
9. Coulson, S.: Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction. Cambridge University Press, Cambridge (2006)
10. Diaconescu, R.: Extra Theory Morphisms for Institutions: Logical Semantics for Multi-Paradigm Languages. Applied Categorical Structures **6**(4), 427–453 (1998)
11. 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)
12. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books, New York (2002)
13. Gentner, D.: Structure-mapping: a theoretical framework for analogy. Cognitive Science **7**(2), 155–170 (1983)
14. 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)

15. Gómez-Ramírez, D.: Conceptual Blending as a Creative Meta-Generator of Mathematical Concepts: Prime Ideals and Dedekind Domains as a Blend. In: T.R. Besold, K.U. Kühnberger, M. Schorlemmer, A. Smaill (eds.) Proceedings of the 4th International Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI), *Publications of the Institute of Cognitive Science*, vol. 02-2015, pp. 1–11. Institute of Cognitive Science (2015)
16. 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)
17. Lakoff, G., Núñez, R.: Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being. Basic Books, New York (2000)
18. 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)
19. 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)
20. MacNish, C.K., Antoniou, G.: Specification Morphisms for Nonmonotonic Knowledge Systems. In: A. Sattar (ed.) Knowledge Representation and Reasoning, *Lecture Notes in Computer Science*, vol. 1342, pp. 246–254. Springer (2005)
21. 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)
22. Martinez, M., Krumnack, U., Smaill, A., Besold, T.R., Abdel-Fattah, A., Schmidt, M., Gust, H., Kühnberger, K.U., Guhe, M., Pease, A.: Algorithmic aspects of theory blending. Proceedings of the 12th International Conference on Artificial Intelligence and Symbolic Computation (AISC), G. Aranda-Corral, F. Martın-Mateos, and J. Calmet, Eds., LNAI, Springer (2014)
23. 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)
24. Mossakowski, T., Maeder, C., Codescu, M.: Hets user guide version 0.99. `http://www.informatik.uni-bremen.de/agbkb/forschung/formal_methods/CoFI/hets/UserGuide.pdf` (2014)
25. Pereira, F.C.: Creativity and AI: A Conceptual Blending Approach. Applications of Cognitive Linguistics (ACL). Mouton de Gruyter, Berlin (2007)
26. Plotkin, G.D.: A Note on Inductive Generalization. Machine Intelligence **5**, 153–163 (1970)
27. Schiralli, M., Sinclair, N.: A constructive response to "where mathematics comes from". Educational Studies in Mathematics **52**, 79–91 (2003)
28. 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)
29. 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)
30. 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)
31. Veale, T., O'Donoghue, D.: Computation and blending. Computational Linguistics **11**(3–4), 253–282 (2000). special issue on Conceptual Blending
32. 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)