City Research Online

City, University of London Institutional Repository

Algorithmic Aspects of Theory Blending

M. Martinez[1], U. Krumnack[2], A. Smaill[3], T. R. Besold[2], A. Abdel-Fattah[2,4], M. Schmidt[2], H. Gust[2], K.-U. Kühnberger[2], M. Guhe[3], and A. Pease[5]

[1] Department of Mathematics, Universidad de los Andes, Bogotá
[2] Institute of Cognitive Science, University of Osnabrück, Osnabrück
[3] School of Informatics, University of Edinburgh, Edinburgh
[4] Computer Science Division, Faculty of Science, Ain Shams University, Cairo
[5] School of Computing, University of Dundee, Dundee

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. This paper describes a logic-based framework which allows a formal treatment of theory blending, discusses algorithmic aspects of blending within the framework, and provides an illustrating worked out example from mathematics.

1 Introduction

Since its introduction, the theoretical framework of Conceptual Blending (CB) has gained popularity as alleged submechanism of several complex high-level cognitive capacities, such as counterfactual reasoning, analogy, and metaphor [2]. While there is a growing body of work trying to conceptually relate CB to several facilities at the core of cognition, there currently are very few (if any) fully worked out formal or algorithmic accounts. Still, if only some of the assumptions made about the importance of blending mechanisms within human cognition and intelligence turn out to be reliable, a complete and implementable formalization of CB and its defining characteristics would promise to trigger significant development in artificial intelligence.

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 [3].
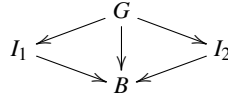
Fig. 1. Goguen's version of concept blending (cf. [3])

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 (set of axioms) in a formal language. We'll call the nodes "spaces", so to avoid terms with strong semantic 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 a 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 [3] and linked to earlier work on computational aspects of blending in cognitive linguistics (see, e.g., [11]), 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 blend is better than other candidate blends. This question has lead to the formulation of various competing optimality principles in cognitive linguistics (cf. [2]). 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 [3], 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 *first-order* theories, in order to obtain the generalization spaces G. Accordingly, our presentation here will be restricted to CB over first-order theories. The paper is structured as follows: we first introduce the formal framework we use to model blending processes, and then propose our algorithmic description of blending. As proof of concept, along the paper we present a worked out example from mathematics. The paper finishes with some concluding remarks, a review of related work, and an outlook for future research.

2 Our Framework

Our approach is based on the *Heuristic-Driven Theory Projection* (HDTP, [10]), a framework for computing analogical relations between two input spaces presented as axiomatizations in some many-sorted first-order languages. HDTP proceeds in two phases: 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 mapped to the target space to establish new hypotheses (Figure 2). For our current purposes we will only need the mapping mechanism and replace the transfer phase by a new blending algorithm, so instead of talking about source and target spaces, from now on we will refer to the input spaces as the 'left' and 'right' spaces (L and R). This convention is meant to be merely a mnemonic relating to our diagrams and not an indication that one space has priority over the other (since we don't need transfer anymore).

$$\textit{Generalization (G)}$$

Source (L) $\cdots\cdots\cdots\xrightarrow{\textit{analogical transfer}}$ *Target (R)*
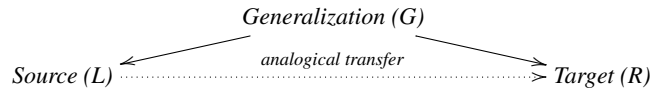
Fig. 2. HDTP's overall approach to creating analogies (cf. [10]).

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 [9] 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 and the only susbtitutions on E that yield anti-unifiers of A and B act as trivial renamings of the variables in E. As it happens, first-order anti-unification (where only first-order substitutions are allowed) is not powerful enough to produce the generalizations needed in HDTP, so a special form of *higher-order* anti-unification is used where, under certain conditions, symbols of relation and function can also be included in the domain of substitutions (see [10] 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*.

Example 1. We will use a working example in this paper 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 $x <_i y$ abbreviates $\neg(y \leq_i x)$. The table also shows a generalization theory G over the signature is $\{a, \leq, +\}$. G 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'll get the first six L-axioms and the first six R-axioms, respectively, which are the covered formulas in this example.

In HDTP, any 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

Table 1. 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.

Axiomatization L		Axiomatization R		Generalization G	
$x \leq_L x$	(L1)	$x \leq_R x$	(R1)	$x \leq x$	(G1)
$x \leq_L y \wedge y \leq_L z \to x \leq_L z$	(L2)	$x \leq_R y \wedge y \leq_R z \to x \leq_R z$	(R2)	$x \leq y \wedge y \leq z \to x \leq z$	(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)		
$x \leq_L y \wedge y \leq_L x +_L 1 \to y = x \vee y = x +_L 1$	(L8)	$x <_R y \to \exists z : x <_R z \wedge z <_R y$	(R8)		

from both spaces. Thus, the blending process can build on the generalization and substitutions provided by the analogy engine, and analogy can be considered a special case of blending.

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 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 [6]. The other extreme (right side of Figure 3) occurs

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

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 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 G, the theories L and R can be split into their (non-empty) covered parts L^+ and R^+ and uncovered parts L^- and R^-. The covered parts are fully analogical, i.e. basically isomorphic, and make up the core of the 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

---

[6] 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 don't yield a good analogical relation (cf. [10, pp. 258]). Thus, in some cases, two formally different but semantically equivalent axiomatizations may not result in an empty generalization.

propose the following two optimality principles: IP renders a version of the Web and Topology principles formulated in the introduction, while CP supports the Unpacking Principle.

> 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).

3 Theory Blending Algorithm

Now we tackle the problem of algorithmically finding a list of *optimal* blends, given two input theories L and R over first-order signatures Σ_L, Σ_R, respectively. A blend is *optimal* if it is consistent and as maximally compressed and informative as possible. An unconstrained way to do this leads to an explosion of possibilities to be tried, so good heuristics are needed in order to choose which possibilities to test first. We propose to proceed according to the following general steps:

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[7]. As an example, see Table 1 and Example 1.

2. Identification: Build the blend signature Σ_B by taking the 'union' of Σ_L and Σ_R and collapsing each pair of symbols aligned by G to only one of them. Regardless of how the collapsing is done, at the end the algorithm will produce the same blends, modulo partial renamings of identified symbols[8]. In what follows, we will simply choose the symbol from Σ_R when collapsing a pair. Thus, for the case of Table 1, Σ_B will coincide with Σ_R, since no symbol in Σ_L is uncovered by the left substitution.

3. Blending: Construct the set of all formulae over Σ_B that might be part of a blend. This will consist of every formula in R^+, the covered part of R, plus every formula in the uncovered parts of R and L, $Ax = Tr(L^-) \cup R^-$. Here Tr is the (partial) translation function that maps symbols from Σ_L to corresponding symbols from Σ_R according to the generalization G, so ensuring that all formulas of Ax are build over signature Σ_B. The set Ax corresponding to the example in Table 1 is listed in the leftmost column of Table 2, which also shows all the candidate blends for this particular generalization G.

Back to the general setting, the set $R^+ \cup Ax$ would be the ideal blend, but it might be inconsistent. So in this step we also compute the set MaxCon of maximal consistent blends B such that $R^+ \subseteq B \subseteq R^+ \cup Ax$. For the running example, this involves exploring the lattice of theories depicted in Figure 4.

---

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

[8] The algorithm might in principle be extended by producing for each discovered optimal blend all of its "mirror" blends, obtained by renamings.

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 $R^+ \cup Ax$ that extend R^+.

4. Relaxation: Reduce the set of symbols covered by the generalization by shrinking G (some simple heuristics for this step are given below). Return to step 2.

Now we discuss how steps 3 and 4 can be implemented (steps 1 and 2 are obtained from HDTP). We use a simple procedure COMPUTEBLENDS which, besides the sets R^+ and Ax introduced above, needs a list Init of initial blend candidates (so each element of Init extends R^+). Init must have the property that every possible blend based on the current generalization 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^+ (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.

proc COMPUTEBLENDS(R^+, Ax, Init, direction)
    global MaxCon := $\emptyset$; **global** MinInc := $\emptyset$
    foreach $T \in$ Init **do** EXPLORE(R^+, Ax, T, direction) **end foreach**
end proc

The first thing to do is to initialize as empty two global sets MaxCon and MinInc that will keep at all times during the search the largest consistent theories and the smallest inconsistent theories 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 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. they will be subsumed by some theory in MaxCon if they are consistent, and they will subsume some theory from MinInc if they are inconsistent (cf. Lemma 1 below). When the loop ends, MaxCon determines precisely the optimal blends.

proc EXPLORE(R^+, Ax, T, direction)

    if $T \notin \downarrow$MaxCon $\cup \uparrow$MinInc **then**
        if T is consistent **then** MaxCon := $\{T\} \cup \{M \in$ MaxCon $\mid M \nsubseteq T\}$
        else MinInc := $\{T\} \cup \{M \in$ MinInc $\mid T \nsubseteq M\}$ **endif**
    endif
    if $T \in \downarrow$MaxCon **and** (direction $\in \{up, both\}$) **then**
        foreach $Axiom \in (Ax \setminus T)$ **do** EXPLORE(R^+, Ax, $T \cup \{Axiom\}$, **up**) **end foreach**
    else if $T \in \uparrow$MinInc **and** (direction $\in \{down, both\}$) **then**
        foreach $Axiom \in T \setminus R^+$ **do** EXPLORE(R^+, Ax, $T \setminus \{Axiom\}$, **down**) **end foreach**
    endif
end proc

Here, $\uparrow C$ denotes the set of theories that contain some theory from C and $\downarrow C$ denotes the set of theories that are contained in some theory from C; $\updownarrow C$ is $\uparrow C \cup \downarrow C$. As first step in EXPLORE, if T is not yet classified by MaxCon or MinInc, consistency of T is checked and MaxCon or MinInc are updated accordingly. In any case, if T is consistent

(inconsistent), a recursive upwards (downwards) search towards extensions (subsets) of T is initiated. These upward and downward searches are performed unless the direction parameter prohibits them. The calls to EXPLORE made when working with the first, strongest generalization 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)[9].

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

Proposition 1. *The following pre- and post conditions hold true of the operation of* EXPLORE *(R^+, Ax, T, direction), 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 executing EXPLORE*, then the same holds afterwards.*
(3) If a theory blend B is classified correctly by MaxCon and MinInc before executing EXPLORE*, then the same holds after calling* EXPLORE*.*
(4) If direction = up and MaxCon and MinInc classify correctly before executing EX-PLORE*, then $\uparrow T$ is classified correctly by MaxCon and MinInc after calling* EXPLORE*.*
(5) If direction = up and MaxCon and MinInc classify correctly before executing EX-PLORE*, then $\downarrow T$ is classified correctly by MaxCon and MinInc after calling* EXPLORE*.*
(6) If direction = both and MaxCon and MinInc classify correctly before executing EX-PLORE*, then $\updownarrow T$ is classified correctly by MaxCon and MinInc after calling* 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^+ and $R^+ \cup Ax$ 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 it will become $\{T\} \cup \{M \in MaxCon | M \not\subset T\}$. 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. (4) We proceed by induction on the cardinality of $Ax \setminus T$. 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 $Ax \setminus T$, a call EXPLORE($R^+, Ax, T \cup \{A\}$, **up**) 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 $T \setminus R^+$. (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. $\square$

---

[9] 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.

As our framework stands, the evaluation of blends in Step 3 and the decision to stop or continue with a relaxation, is a mandatory interactive step where the user decides. As for the relaxation step, if needed, it is important to find a good weakening of G a good set Init with which to continue to step 2. 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: (1) only one old symbol mapping is dropped, and (2) the fewest number of axioms become uncovered under the new generalization.

In any case, once a weakened generalization $\hat{G}$ has been fixed, the previously found MaxCon and MinInc sets are used to compute an appropriate new Init set, as follows. Let Tr and $\hat{Tr}$ 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 $Tr(\phi)$ in R^- by $\hat{Tr}(\phi)$. This new Init is good in that every optimal blend for the weakened generalization will be an extension of one the Init elements. This is why the exploration, after some relaxation has been made, can be constrained to be upwards only.

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[10], 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 [7] 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.

4   Worked Example

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 natural numbers (starting from 1) and L 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 4 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, Init:= $\{T1, T3, T7, T4\}$.

---

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

Table 2. Formulae *L7t* and *L8t* result from transferring the uncovered formulae of axiomatization *L*, according to generalization *G*. The table shows some of the theories in the search space of possible blends. Maximal consistent theories are starred.

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

The sets MaxCon and MinInc are initialized as empty and we start to explore the initial theories. The first is *T*1, which is inconsistent:

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

The two last lines are clearly contradictory. The algorithm orders to add *T*1 to MinInc. However, knowing that the inconsistency arises from only the axioms *R*1, *R*7, and *L*7*t*, it is better to add the smaller *T*5 to MinInc than adding *T*1 itself. Thus, MinInc:= {*T*5}.

Now, as the algorithm prescribes, we recursively explore (downwards) every theory obtained from *T*1 by deleting one axiom. These theories are *TR*, *T*2, and *T*5: *TR* is consistent and *T*5 ⊄ *TR*, so MaxCon := {*TR*}; *T*2 is consistent, not contained in *TR*, and does not extend *T*5, then we update MaxCon := {*TR*, *T*2}; and *T*5 extends the only member of MinInc, so we do nothing. This ends the analysis of *T*1.
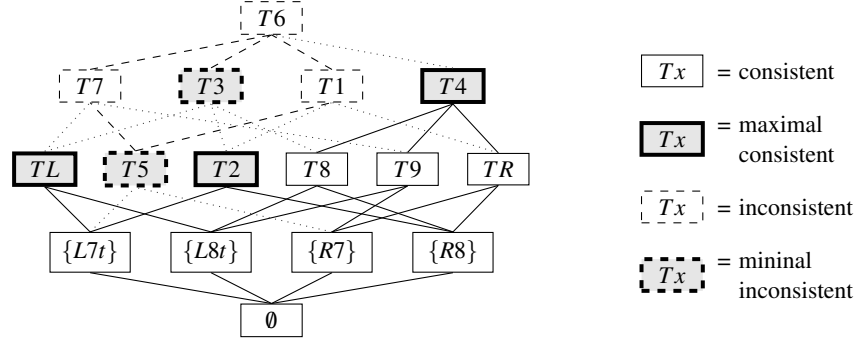
Fig. 4. A lattice of the 'blends' that appear in the given example.

The second initial theory is *T*3. This theory is not a subset of *TR* or *T*2, and does not extend *T*5. In addition it is inconsistent, as shown by the third and last lines of the following proof, which uses all the axioms of *T*3 not covered by the generalization.

$$\neg(x +_R 0 \leq x) \tag{L7t}$$

$$\neg(x +_R 0 \leq x) \rightarrow \exists z : (x <_R z \wedge z <_R x +_R 0) \tag{R8}$$

$$x <_R z \wedge z <_R x +_R 0 \tag{FOL}$$

$$\neg(z \leq_R x) \wedge \neg(x + 0 \leq_R z) \tag{Def. $\leq_R$}$$

$$x \leq_R z \wedge z \leq_R x +_R 0 \tag{FOL + R3}$$

$$z = x \vee z = x +_R 0 \tag{MP with L8t}$$

$$z \leq_R x \vee x +_R 0 \leq_R x \tag{FOL + R1 + Def. $\leq_R$}$$

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 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 $T j$ from the first stage. We call them $T j0$, $T j1$, $T j2$, or $T j3$ depending on which subset of $\{R4, L4tt\}$ they contain: $T j0$ includes no element from $\{R4, L4tt\}$, $R j1$ includes only $L4tt$, $R j2$ includes only $R4$, and $R j3$ 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 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

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
$(R1)-(R3),(R5),(R6)$		X	X	X	X	X	X	X	X	X	X	X
$0 \leq_R x$	$(R4)$				X	X			X	X	X	X
$x +_R 0 = x$	$(R7)$		X	X	X	X	X	X	X	X	X	X
$x <_R y \to \exists z : (x <_R z \land z <_R y)$	$(R8)$	X					X	X	X	X	X	
$1 \leq_R x$	$(L4tt)$			X		X		X		X		
$\neg(x +_R 1 \leq_R x)$	$(L7tt)$	X	X	X	X	X	X	X	X	X	X	X
$x \leq_R y \land y \leq_R x+1 \to y = x \lor y = x +_R 1$	$(L8tt)$	X								X	X	X
Consistent:		N	Y	N	Y	N	Y	N	Y*	N	N	Y*

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 MinInc := $\{T30, T51\}$.
2. $T10$ is consistent and extends $T50$. Set 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 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 $T30 \in$ MinInc.
3. $T70 = T50 \cup \{L8tt\}$ is consistent. So we update MaxCon:= $\{T12, T70\}$, and explore the minimal extensions of $T70$. They are: $T60$ (which extends $T30 \in MinInc$), $T71$ (which extends $T51 \in MinInc$), and $T72$ (maximal consistent). After these explorations, MaxCon:= $\{T12, T72\}$, and MinInc:= $\{T30, T51\}$.
4. $T52$ is a subset of $T12 \in$ 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.

5 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 [2]'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 [8], as there mappings "do not have to rely on similarity: they can present conflicts that are striking, surprising or even incongruous" [8, 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.

Other algorithmic accounts are given, for instance, in [8], where the CB mechanism uses a parallel search engine based on genetic algorithms, or in [4], sketching the blending of logical theories within a distributed ontology setup. Further work on CB is contained in [6] where the authors present a rule-based system for counterfactual reasoning in natural language. These examples are mostly addressing problems from linguistics or philosophy, but our interest lies in particular in the blending of mathematical theories, as a means of understanding certain developments in the history of mathematics, as described by [1], and as part of general mathematical cognition, as suggested by [5].

References

1. Alexander, J.: Blending in Mathematics. Semiotica 2011(187), 1–48 (2011)
2. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books, New York (2002)
3. Goguen, J.: Mathematical models of cognitive space and time. In: Andler, D., Ogawa, Y., Okada, M., Watanabe, S. (eds.) Reasoning and Cognition: Proc. of the Interdisciplinary Conference on Reasoning and Cognition. pp. 125–128. Keio University Press (2006)
4. 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)
5. Lakoff, G., Núñez, R.: Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being. Basic Books, New York (2000)
6. 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)
7. Manna, Z., Zarba, C.: Combining Decision Procedures. In: Formal Methods at the Crossroads: from Panacea to Foundational Support, pp. 381–422. LNCS 2787, Springer (2003)
8. Pereira, F.C.: Creativity and AI: A Conceptual Blending Approach. Applications of Cognitive Linguistics (ACL), Mouton de Gruyter, Berlin (2007)
9. Plotkin, G.D.: A Note on Inductive Generalization. Machine Intelligence 5, 153–163 (1970)
10. 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)
11. Veale, T., O'Donoghue, D.: Computation and blending. Computational Linguistics 11(3–4), 253–282 (2000), special issue on Conceptual Blending