

A computational account of conceptual blending in basic mathematics

Action editor: Cleotilde Gonzalez

Markus Guhe^{a,*}, Alison Pease^a, Alan Smaill^a, Maricarmen Martinez^{b,c}, Martin Schmidt^b,
Helmar Gust^b, Kai-Uwe Kühnberger^b, Ulf Krumnack^b

^a University of Edinburgh, School of Informatics, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, Scotland, United Kingdom

^b University of Osnabrück, Institute of Cognitive Science, Albrechtstr. 28, 49076 Osnabrück, Germany

^c Universidad de los Andes, Department of Mathematics, Cra 1a 18A-10, Bogotá, Colombia

Received 2 May 2010; accepted 16 September 2010

Available online 3 February 2011

Abstract

We present an account of a process by which different conceptualisations of number can be blended together to form new conceptualisations via recognition of common features, and judicious combination of their distinctive features. The accounts of number are based on Lakoff and Núñez's cognitively-based grounding metaphors for arithmetic. The approach incorporates elements of analogical inference into a generalised framework of conceptual blending, using some ideas from the work of Goguen. The ideas are worked out using *Heuristic-Driven Theory Projection* (HDTP, a method based on higher-order anti-unification). HDTP provides generalisations between domains, giving a crucial step in the process of finding commonalities between theories. In addition to generalisations, HDTP can also transfer concepts from one domain to another, allowing the construction of new conceptual blends. Alongside the methods by which conceptual blends may be constructed, we provide heuristics to guide this process.

© 2011 Published by Elsevier B.V.

Keywords: Mathematical cognition; Metaphor; Mathematical reasoning; Analogy; Anti-unification; Conceptual blending; HDTP

1. Introduction

Conceptual blending plays a central role in the development of mathematical ideas and in the discovery of new mathematical concepts. For us, this is particularly relevant, because one aim of our research is to include cognitive processing principles in the automated discovery of mathematical ideas. In this paper, we look at an example which involves the grounding of mathematical notions in basic cognitive capacities, as well as the relationships between these capacities and more abstract arithmetic conceptualisations.

Before we go into the details of these issues, however, we will first present some of the context of this work from the viewpoints of the philosophy of mathematics and of the role that analogy and metaphor play in mathematical cognition. A new focus in the history and philosophy of mathematics on mathematical practice has led to interest in the relationships that appear when mathematical concepts, definitions, theories and techniques are evolving. Lakatos's (1976) influential work already emphasised the dialectical nature of concept change and the interaction between concepts, conjectures, counterexamples and proof in the development of each. More recently, in his work on the philosophy of mathematical practice, Mancosu (2008) identifies the development of mathematical concepts as one of eight topics with which philosophers of mathematics should concern themselves. Tappenden (2008, p. 257) argues that 'the discovery of a proper definition is rightly regarded in practice as a significant contribution to mathematical knowledge' and quotes Harris: 'The progress of

* Corresponding author.

E-mail addresses: m.guhe@ed.ac.uk (M. Guhe), a.pease@ed.ac.uk (A. Pease), a.smaill@ed.ac.uk (A. Smaill), mmartine@uos.de (M. Martinez), martisch@uos.de (M. Schmidt), hgust@uos.de (H. Gust), kkuehnbe@uos.de (K.-U. Kühnberger), krumnack@uos.de (U. Krumnack).

algebraic geometry is reflected as much in its definitions as in its theorems' (Harris, 1992, p. 99, quoted in Tappenden, 2008, p. 256). Tappenden also urges applying to mathematics the principle by Arnauld and Nicole (1683) that 'nothing is more important in science than classifying and defining well'. As these quotations suggest, mathematical concepts are not born in ready-to-use Platonic fashion, they evolve over time according to need, application, simplicity, beauty and a host of other difficult-to-identify criteria.

Corfield (2003, p. 21) holds that 'No mathematical concept has reached definitive form – everything is open to reinterpretation', thus advocating a historical approach to analysing concept formation. In the history of mathematics much work traces the historical development of particular mathematical concepts, such as ZERO (Kaplan, 2000), SYMMETRY (Stewart, 2007) and PRIME NUMBERS (Gray, 1992). Some of these historical developments are candidates for analysis in terms of conceptual blending; indeed Alexander (2009) provides insight into how these events can be seen in such terms.

At the centre of the cognitive account of how people make mathematical discoveries are processes such as analogy, metaphor and conceptual blending. This view is somewhat different from the traditional view that deduction is the main means of mathematical reasoning, or even from much more recent (and controversial) views that scientific induction¹ (Polya, 1962) and abductive reasoning (Arzarello, Andriano, Olivero, & Robutti, 1998) play an important role. While deductive reasoning plays an integral part in formulating a mathematical argument, in particular presenting a proof in a mathematical paper, analogical reasoning and other cognitive processes like conceptual blending can (along with scientific induction and abduction) help to shed light on how mathematicians create mathematical concepts, conjectures and initial proof attempts. This is worth pointing out, because while the role of analogy, for example, is more or less uncontroversial in cognitive science, many mathematicians find it hard to acknowledge the central role of non-formal means in the progress of mathematics. A reason for this may be that mathematics is often attributed a special status – physicists, for example, seem less reluctant to take this view, e.g. Maxwell explicitly pointed out the role that analogical reasoning played in his discoveries (Nersessian, 2008). It should be noted, however, that Polya (1954, p. 17) observed that 'analogy seems to have a share in all mathematical discoveries, but in some it has the lion's share'. He provides historical examples, where analogy played a key role, such as the discovery by Euler of the conjectured solution to the Basel problem $\sum \frac{1}{n^2}$ (the problem of finding the sum of the reciprocals of the squares of the natural numbers) as $\frac{\pi^2}{6}$ (although Euler later used deductive methods to prove his conjecture).

When scientists attempt to account for a newly observed phenomenon, they often construct analogies to existing scientific theories. A famous example is attributed to Rutherford, namely the analogy between the structure of the solar system and the structure of the atom.²

Cognitive accounts of analogy are often given in the form of *structure mapping theory*, proposed by Gentner (1983); see also (Gentner & Markman, 1997): creating an analogy between two knowledge representations (domains) consists in finding structural correspondences between the domains. For example, in a simplified version of the solar system–atom analogy, the SUN is mapped onto the NUCLEUS and a relation like REVOLVES-AROUND (which holds between PLANET and SUN) is mapped onto REVOLVES-AROUND in the atom domain. Structure mapping emphasises that to create an analogy, correspondences between (higher-level) relations have to be established, e.g. between the two REVOLVES-AROUND relations, or the fact that the SUN/NUCLEUS is in an ATTRACTS relation with the PLANET/ELECTRON.

For the following, we will assume that the general cognitive mechanism for establishing metaphors is essentially the same as the one for establishing analogies (Gentner, Bowdle, Wolff, & Boronat, 2001). In particular, we follow the tradition originating with Lakoff and Johnson (1980) and take a metaphor to be a mapping between *conceptual spaces* (also often called *mental spaces*, cf. Fauconnier & Turner, 2002), which has conceptually strong similarities to analogy making in the tradition of the structure mapping approach.

The conceptual spaces we will start from in this paper are taken from the four basic metaphors of arithmetic proposed by Lakoff and Núñez (2000). These basic metaphors are OBJECT COLLECTION, OBJECT CONSTRUCTION, MEASURING STICK and MOTION ALONG A PATH. According to Lakoff and Núñez, these four metaphors play a role in grounding the human conception of arithmetic in everyday experiences. For example, according to the OBJECT COLLECTION metaphor, our understanding of addition is supported (among other things) by our understanding of facts such as that if we add a collection of two objects to a collection of three objects, we get a collection of five objects. A central problem of this overall approach, which is often neglected (and we will not elaborate on either), is the question of how a domain or a conceptual space is established. We will assume that knowledge about the solar system is different enough from knowledge about the atomic structure or that time is sufficiently different from space that there is no danger of confusing or mixing them.

In Guhe et al. (2010) we have already shown how new mathematical knowledge can be generated from the mentioned basic domains by a process of generalisation with *Heuristic-Driven Theory Projection* (HDTP), an algorithm-

¹ Scientific induction is contrasted with mathematical induction, which is deductive.

² Versions of the atom reminiscent of the solar system existed before Rutherford's and colleagues' experimental results in 1911, concepts like inertia and forces of attraction being at the centre of the analogy.

mic system that computes generalisations of given source and target domains (Schwering, Krumnack, Kühnberger, & Gust, 2009). HDTP is based on anti-unification (Plotkin, 1970) and allows a mapping between source and target via the computed generalisation. However, metaphorical mappings between conceptual spaces are a special case of *conceptual blending* (Fauconnier & Turner, 2002). In this paper we argue that metaphorical mappings alone are not sufficient to account for how mathematical concepts are created. Instead, the more comprehensive account of conceptual blending is required.

Like the basic conception of metaphors, conceptual blends are based on interrelated conceptual spaces. However, conceptual blends go beyond metaphorical mappings in that they create new conceptual spaces (*blend spaces*) that combine the original conceptual spaces in novel ways and that also make it possible to create new concepts by conceptual combination. For example, in the well-documented TIME IS SPACE metaphor (Fauconnier & Turner, 2002, 2008) a standard mapping consists in mapping the progression of time to moving through space. However, only a conceptual blend provides the necessary representation for a conceptualisation visible in an expression like *time passes*, which requires one to conceptualise the OBSERVER to be static and TIME AS MOVING past him/her. In other words, only by having created a new conceptual space – the blend space – has it become possible to recombine concepts from both source domains (TIME and SPACE) in this novel way.

In Fauconnier and Turner's account, a mapping between conceptual spaces is maintained once it has been established. This means that all conceptual spaces will be updated when a change occurs in one space. While it is unclear whether this claim is tenable in a cognitive account (because it is computationally potentially very costly), we will make this assumption here too. The rationale underlying the assumption that multiple conceptual spaces exist simultaneously instead of being merged into one big space is that despite the additional knowledge gained from, for example, the blend TIME IS SPACE, the knowledge about SPACE will remain (*mutatis mutantis*) the same. That is, even if additional knowledge about TIME IS SPACE is available, this knowledge is usually not needed or considered by a (cognitive) system when using spatial knowledge.

In this paper, we do not make behavioural predictions or suggest particular empirical studies, but we acknowledge their importance for understanding conceptual blending in mathematical thinking. Here we are instead concerned with (1) giving a formal, computational account of conceptual blending and (2) exploring its role in the structures required for conceptualisations used in mathematical thinking. The main reason for not considering empirical data is the difficulty in obtaining such data for creative scientific thinking, see, for example, the discussions in Nersessian (2008). In what sense, then, is our model cognitive? We do not consider cognitive models to be constrained to models that replicate a particular experimental result but rather see us

in the tradition of Alan Newell's proposal to create unified theories of cognition (Newell, 1990). The theories we are unifying here are analogical thinking in the form of structure mapping, generalisation in the form of anti-unification and conceptual blending in the form of the theory of Institutions. All three theories model high-level cognitive processes central to complex cognition. By integrating them, we better understand the role of these mechanisms for abstract (mathematical) thinking and the interplay between them. Furthermore, as the mathematical structures that our model creates are the result of human thinking, we consider our model to be a suitable first account of these processes in complex cognition.

It should also be noted that we are not suggesting that humans apply exactly those blending strategies described in the following rather than alternative ones. Our main purpose here is to demonstrate that conceptual blending is a vital aspect of mathematical creativity and to provide a proof of concept for a computational model of this cognitive process.

2. Three steps of mathematical discovery via conceptual blending

Our computational approach portrays the process of mathematical discovery via blending as consisting of three steps: exploration, recognition/goal formulation, and discovery, which we will outline in this section. Our running example will take as input formalisations of the OBJECT CONSTRUCTION, MEASURING STICK and MOTION ALONG A PATH grounding domains for arithmetic. The formalisations (first order theories) do not necessarily stand for the grounding domains themselves but for understandings of them, 'naive precursors' of the abstract number domains. For instance, in the OBJECT CONSTRUCTION domain any object consisting of three unit parts is a precursor of the abstract number three.³ Our running example illustrates how a system can compute, out of the basic domains, a new notion of number (here a new theory) that approximates our notion of rational numbers. In particular, the created domain will have three properties:

1. There is exactly one precursor of each natural number. On many occasions we will refer to the precursor of a number as if it were a number itself.
2. The elements of the domain are linearly ordered. The domain captures the idea of numbers being locations (points) in a linear path.
3. The domain includes precursors of rational numbers, which can be described in the domain as being *fractions* of other objects. Fractions that should be equivalent

³ Lakoff and Núñez's stance is anti-Platonist. We do not take any philosophical position on the ontological status of numbers here, but as our example is based on Lakoff and Núñez's ideas, we made an effort to choose terminology that does not suggest the existence of a Platonic realm of numbers.

according to abstract arithmetic are indeed the same object in this domain. For example, there is only one object that is *one half of 1* and this is the same unique object that also is *two fourths of 1* in the domain.

Lakoff and Núñez's choice of their four metaphors as the basic ones for grounding arithmetic, is based on an analysis of linguistic expressions. Thus, expressions such as *adding onions to the soup* (physically placing an object into a container) or *take a book out of the box* (physically removing an object or substance from a container) suggest that interactions with objects and collections of objects in the physical world are used to construct basic arithmetical concepts and operations and is reflected in the way they are described linguistically. Lakoff and Núñez capture this idea in their ARITHMETIC IS OBJECT COLLECTION metaphor. Besides object collections, they also propose the domains of OBJECT CONSTRUCTION, MEASURING STICK and MOTION ALONG A PATH as grounding domains for arithmetic. The last three domains are the ones we will use in our working example. We will explain the metaphors in more detail in Section 4, but intuitively they consist in grounding arithmetic in the following domains:

- constructed objects like a tower of toy blocks, where the constructed objects correspond to numbers and addition corresponds to building a new constructed object out of two given constructed objects;
- using a MEASURING STICK to mark out different lengths, where the constructed lengths correspond to numbers and addition corresponds to placing two such lengths one after another;
- motion along a path, where numbers correspond to certain point locations on a path and addition corresponds to moving further away from the origin of the path.

Fractions are a concept mainly motivated by the OBJECT CONSTRUCTION domain, where $5/2$ in arithmetic corresponds to a constructed object built by splitting several unit objects into pairs of half-sized objects and gathering 5 of the smaller objects into a collection. Rational numbers, on the other hand, are mainly motivated by an enriched version of the MOTION ALONG A PATH domain, where they correspond to point locations in between or at those point locations on a path that are multiples of a unit length. The notion of a unit length is taken from the MEASURING STICK domain.

2.1. Step 1 – exploration

The first step towards making a mathematical discovery is to explore relationships between the available conceptual spaces – the grounding domains in our case. This is done by the analogy making engine *Heuristic-Driven Theory Projection* (HOTP). The results are mappings between the representations of OBJECT CONSTRUCTION, MEASURING STICK and MOTION ALONG A PATH. Some of the mappings contain,

for example, a correspondence between objects (from OBJECT CONSTRUCTION), distances (from MEASURING STICK), and point locations on a path (from MOTION ALONG A PATH). There is also a mapping of the smallest elements of the first two domains. Two aspects of these mappings can be used for creating new knowledge:

- Not all knowledge of the two domains is mapped: parts will be ‘left over’.
- There are entities that cannot be mapped in spite of sharing similar properties. For example, as described in Section 4.2, the smallest element of the OBJECT CONSTRUCTION domain cannot be mapped to the smallest element of the MOTION ALONG A PATH domain because the former element behaves as the 1 of arithmetic in that adding it to another object yields a strictly larger constructed object, while the smallest element of the MOTION ALONG A PATH behaves as the zero of arithmetic, being an identity for the ‘addition’ operation of the domain.

2.2. Step 2 – recognition and goal formulation

The mappings of step 1 result in recognising, among other things, that multiple fractions in the OBJECT CONSTRUCTION domain, e.g. $1/2$ and $2/4$, map onto the same point on a path, namely the point half-way in between the origin and the unit location. As already mentioned, when we talk, for example, about the fraction $1/4$ in the OBJECT CONSTRUCTION domain we are not referring to the abstract number but to any object that is a fourth part of a unit size object. Based on this observation, the system can formulate a goal to discover whether there is a regularity with which these fractions are related, for example the goal can be to try to find out how the existing definition of equality in OBJECT CONSTRUCTION (two fractions a/b and c/d are equal if $a = c$ and $b = d$) must be changed to accommodate this finding.

Another conjecture based on the mappings of step 1 uses the observation that two of the domains have smallest elements. Based on this mapping, the system can formulate various goals, including:

1. Transfer 0 from MOTION ALONG A PATH to MEASURING STICK or to OBJECT CONSTRUCTION because it is a ‘new’ number.
2. Hypothesise that the UNIT DISTANCE from the MEASURING STICK, and the ORIGIN from MOTION ALONG A PATH are equal and formulate a goal to show that this is indeed the case.
3. Make a conjecture that the SMALLEST OBJECTS from OBJECT CONSTRUCTION and ORIGIN from MOTION ALONG A PATH are equal and formulate a goal to show that this is indeed the case.
4. Hypothesise that the UNIT DISTANCE from MEASURING STICK and ORIGIN from MOTION ALONG A PATH are equal and formulate a goal to show that this is indeed the case.

5. Since there is a mapping between MOTION ALONG A PATH and MEASURING STICK, try to create a domain that is more abstract than the two grounding ones in that objects of the new domain have the properties of distances and of points in a path.

2.3. Step 3 – discoveries

Discoveries that can arise from the previous goals and conjectures include a revised notion of equality for fractions ($1/2 = 2/4$) or the addition of a new object that behaves as zero (and that we will call 0) to domains that lacked such an element.

1. Because multiple fractions can map onto the same point, they can be considered to be the same (rational) number. If the system can work out the regularity, i.e. that such fractions are related in a principled manner (by having a common multiplier/divisor for counter and denominator), it discovers a new definition of equality between fractions.
2. After 0 has been transferred from MOTION ALONG A PATH to OBJECT CONSTRUCTION, the system will note that the smallest entities of the two domains are not the same entity. It will also note the limited applicability of the new concept (ZERO) for fractions, because ZERO cannot be used as denominator. This can, for example, be done by observing that there is no rational number that corresponds to such fractions.

It is important to note that the plausibility of formulating one goal or another depends on the order in which the various mappings and discoveries are made. The status of some of the goals and discoveries listed in this section may vary depending on the particular development: what is a discovery in one history may be directly recognised out of the analogical mappings in another or be formulated as a goal. In Section 4, we will illustrate one way in which such of history of finding mappings, conjecturing and discovery may develop. Our example unfolds in a way such that at some point goal 5 above is formulated. Note that among the ones we mentioned, goal 5 is the one that most explicitly asks for the creation of a conceptual blend.

2.4. Computational realisation

In Section 4 we describe progress in carrying out this programme. The exploration step is carried out automatically using the HDTP engine. For the subsequent steps, the results have been derived by hand. Throughout, various heuristics are employed to make these ideas tractable: HDTP already has associated heuristics, cf. Section 3.3. Heuristics for the subsequent steps are described later in this paper. We are currently working on a full implementation of the proposed extension of HDTP for conceptual blending.

3. Conceptual blending and HDTP

A number of systems has been developed for performing automated concept formation in mathematics, including (Colton, 2002; Epstein, 1988; Fajtlowicz, 1988; Lenat, 1976; Sims & Bresina, 1989). Usually, they explore a domain empirically and construct concepts which describe subsets of the domain they are working in. The techniques they use can be contrasted with those used in concept formation in machine learning, in which a set of positive and negative examples are given in order to automatically construct a concept to cover exactly the positive examples.

Currently, no automated concept formation system in mathematics that we know of uses conceptual blending as a technique; the closest may be Colton's (2002) HR program, which uses a production rule *compose*. This production rule takes two concepts, a primary and a secondary concept, as input. It then 'overlaps' the rows of the primary concept with those of the secondary concept in a way which has been specified by taking a value from a range which has been specified by the user at the beginning of a run. The rule was motivated by the *composition* of two functions; for instance, given functions $f(x)$ and $g(x)$, it was designed to construct the function $f(g(x))$.

In this section, we will look at mechanisms for conceptual blending and how they relate to notions of analogical inference. We will also describe the HDTP system, and some other work that has been done on mechanisms for mathematical concept formation and discovery.

3.1. Conceptual blending and analogy

In the literature, there are various approaches to analogical matching, analogical inference and transfer, for example, Gentner's structure mapping (Gentner & Markman, 1997), Hofstadter's Copycat work (Hofstadter & Mitchell, 1994) or Keane's Incremental Analogy Machine (Keane, 1990). Typically, these approaches regard analogy between two (conceptual) spaces as an asymmetrical relationship between a source and a target. In the structure mapping approach, for example, this is formally reflected by the use of a (partial) mapping in the mathematical sense: structural features in the source domain are mapped to uniquely corresponding features in the target. An alternative, chosen by HDTP, is to allow more general relational correspondences between the given spaces in terms of mappings from a generic space to both of the given ones. Features which are the image of the same common feature in the generic space are then considered to be in structural correspondence. At this level, the formal description is now symmetrical between the spaces, and a feature in the 'source' may correspond to more than one feature in the 'target'.

The notion of a conceptual blend extends this story. A standard example, which is discussed by Goguen (2006), is that of the possible conceptual blends of the concepts HOUSE and BOAT into both BOATHOUSE and HOUSEBOAT. Parts

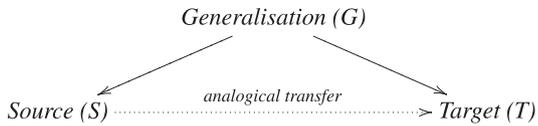


Fig. 1. HDTP's overall approach to creating analogies.

of the conceptual spaces of HOUSE and BOAT can be 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. For our purposes, in this case we note the following:

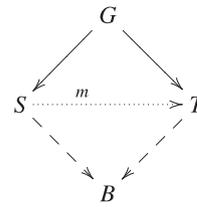
- The blend spaces coexist with the original spaces. Here we agree with Fauconnier and Turner (2008, chap. 2) that in this sort of case we keep in mind the different spaces, and the relationships between them, rather than throwing away one space in favour of the blend. We still want the concepts of HOUSE and BOAT, even as we are aware that a BOATHOUSE has some relation to both HOUSE and BOAT.
- The blend space is not the result of extending one of the given spaces with new features or properties, i.e. it is not a simple enrichment of either of the input spaces. In the mathematical case, where we care about the consistency of the conceptualisations, this means that conceptual blending gives us a way to understand how theories that would simply be inconsistent if crudely combined, can nevertheless be blended by taking appropriately chosen parts of the theories, while respecting common features. Fauconnier and Turner (2002, pp. 242–245) remark on this issue when blending notions of number which are discrete in one case and dense in another.

3.2. Our approach

In the following sections we will develop our ideas in a formal way. Our modelling will be based on *Heuristic-Driven Theory Projection* (HDTP), a framework for analogical reasoning (Schwering et al., 2009). HDTP seems especially suitable for our task, as it provides an explicit generalisation of two domains as a by-product of establishing an analogy. Such a generalisation can be a base for concept creation by abstraction. HDTP creates an analogy in two phases: in the mapping phase, the source and target domains are compared to find structural commonalities, and in this process, a generalised description is created, which subsumes the matching parts of both domains. In the transfer phase, unmatched knowledge in the source domain can be mapped to the target domain to establish new hypotheses, cf. Fig. 1.

For our current purposes, we have to extend this framework to include conceptual blending. Goguen has provided a mathematical framework in which the notion of a

HDTP's view of blending



Goguen's blending diagram

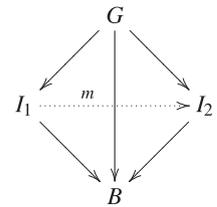


Fig. 2. HDTP's view of conceptual blending compared to Goguen (2006). In the HDTP diagram, the top arrows are substitutions resulting from the computation of the analogical relation m . The dashed arrows symbolise the heuristic-driven construction of the conceptual blend. In Goguen's diagram, the arrows depict morphisms in an appropriate category and the dotted arrow m stands for a cross-domain mapping $I_1 \rightarrow I_2$.

conceptual blend is given a more precise yet flexible characterisation (Goguen, 2006). In this approach, domains⁴ can have both syntactic and semantic components, and mappings from one such domain to another, preserving the conceptual structure, serve as morphisms in a mathematical category. A blend of two given domains I_1 and I_2 is then given by an identification of some shared aspects of I_1 and I_2 and a new domain B , which reflects the shared aspects of I_1 and I_2 while allowing other aspects of I_1 and I_2 to be inherited as well. Operations like analogical transfer and mapping back can also be understood as forming a conceptual blend between two domains. However, in addition to this, the general notion of conceptual blending allows the formation of blends, where the result is not an extension of either given domain but rather takes and rejects selected aspects from both domains.

Goguen's ideas are useful in understanding the computation of blends as developed in this paper. We re-interpret Goguen's theory of conceptual blending as a process of information integration of two domains I_1 and I_2 over a shared domain G , as depicted in Goguen's blending diagram, cf. Fig. 2. Based on a blend B a 'cross space mapping' $m: I_1 \rightarrow I_2$ can be constructed as a reflection of the identifications that are made in B , which reflect identifications inherited from G . The generic space G contains sort (type) information and thereby indicates which individuals should be identified.

This analysis of blending fits HDTP's model of analogy making, although there are some differences: First, the status of the input domains in analogy making is usually not symmetric. Instead, analogies are directed from a (well-known) source domain S towards a (little-known) target domain T . This is not a fundamental difference, but rather indicates properties of the typical examples considered, and the uses made of the analogy or blending, respectively. Second, in HDTP, the generic space G is computed as a generalisation of source and target. Third, the principles for introducing concepts into the target domain by analogical transfer seem to be more restricted than for blending.

⁴ We use the term domain to refer to the source and target. Goguen developed his theory for conceptual spaces.

Despite these differences, the general framework of HDTP can be adapted to these ideas of conceptual blending. The main point, where conceptual blending differs from analogical reasoning is the second phase – the analogical transfer of knowledge from S to T . To turn HDTP into a blending framework, we replace this transfer by a process that results in the creation of new domain, as follows:

1. Compute a common generalisation of the domains S and T , thus setting up relationships between the source and target domains.
2. Create the conceptual blend B by merging knowledge from S and T based on this mapping: in the ideal case, B respects the shared features of S and T (those with common generalisations), and inherits independently the other features of S and T .

In analogical transfer, the operations for modifying the target domain are usually considered to be quite restricted: all that can be done is to enrich the target domain T by knowledge transferred from S , as long as no inconsistencies are introduced. In conceptual blending, however, both domains are merged in a way that common parts are identified. Unmatched parts of the domains will be imported into the blend. Of course, this process may introduce inconsistencies, and there are different ways to deal with them. The two strategies here are to either exclude parts of the inconsistent knowledge from the blend, or to reduce the coverage of the generalisation and thereby the amount of knowledge that is identified in the conceptual blend. We will discuss both strategies in more detail in the worked example.

3.3. Syntactic principles of HDTP

In this section, we provide a short overview of the basic mechanisms of HDTP. A more detailed presentation of these mechanisms can be found in [Schwering et al. \(2009\)](#). HDTP is designed to compute analogical relations and inferences for domains that are represented in a many-sorted first-order logic language. Both, the source and the target domain are given by axiomatisations, i.e. they are represented by finite sets of first-order formulae. The basic idea is to associate (align) pairs of formulae from the two domains in a systematic way, namely by performing anti-unification ([Plotkin, 1970](#)). By computing the anti-unifier of two formulae, these formulae are compared and the least general generalisation that subsumes both formulae is identified.

Fig. 3 depicts some examples of the anti-unification of terms. The anti-instance (generalisation) of two terms is

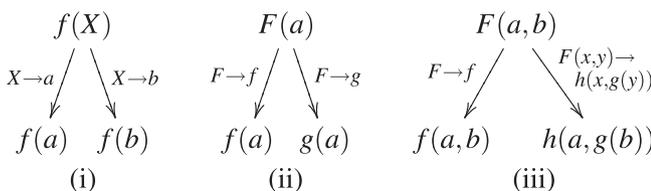


Fig. 3. Anti-unification of terms.

computed by replacing different constants or function symbols by a variable. In example (i) a simple application of first-order anti-unification is sufficient to yield the generalisation $f(X)$. However, the terms in example (ii) differ in the function symbols, i.e. first-order anti-unification fails to detect structural commonalities. To obtain the structural commonality between the two terms, higher-order anti-unification can be used to generalise function symbols to a variable. It is even possible to generalise terms in which common parts are embedded structurally in a different way, as shown in (iii).⁵ Substitutions accompanying the generalised terms are created, which can be used to reconstruct the original terms.

Classical anti-unification of terms is not sufficient to compute commonalities of domains that are represented by first-order logical theories. Therefore, HDTP extends this classical form of anti-unification of terms to formulae and logical theories by iteratively picking pairs of formulae to be generalised from the domains. This process is driven by heuristics. Coherent mappings are preferred over incoherent mappings, i.e. mappings in which substitutions can be reused are considered to be better candidates to induce the analogical relation than mappings with many isolated substitutions. The generalised theory together with its substitutions specifies the analogical relation between source and target. Additional information about the source domain, i.e. formulae of the source domain with no correspondence in the target domain, can be ‘analogically’ transferred to the target by transformation using the already established substitutions.

The generalisation consists of formulae for which there exist instantiations in the source and the target domain. These instantiations are said to be (directly) covered by the generalisation. A domain formula is indirectly covered, if it can be inferred from covered formulae. There may also exist formulae which are uncovered. Such formulae have no correspondence in the other domain and therefore are candidates for transfer in the process of analogical reasoning. In the context of conceptual blending, they contribute domain-specific knowledge to the blend.

4. Worked example

In the following example, we show how a conceptualisation of rational numbers of the form $n/2^k$ (with $n, k \in \mathbb{N}$), close to the one, where rational numbers are points on the real line, can be achieved from some basic grounding domains (in the spirit of Lakoff and Núñez) by establishing metaphorical mappings and blending constructions. Our point of departure are formalised versions of three out of the four basic grounding domains of Lakoff and Núñez: the MEASURING STICK domain, the OBJECT CONSTRUCTION domain, and the MOTION ALONG A PATH domain. We use first

⁵ HDTP uses a restricted form of higher-order anti-unification, that allows us to expand terms by introducing arguments and nested structures as described in [Krumnack, Schwering, Gust, and Kühnberger \(2007\)](#).

order logic with sorts, which adheres closely to the way the domains are presented in Lakoff and Núñez's original account. We do not present all of the embodied, pseudo-arithmetical properties of each domain but only those that are relevant for our purposes here.

We have chosen to formalise the theories we compare within first-order predicate calculus, and we have explicit use of the existential quantifier. The tractability of reasoning over such theories depends on the logical complexity of the theories concerned. We have not aimed for the simplest possible theories here, since our concern is primarily with the interrelationships *between* theories. It is an interesting question to what extent simpler theories are psychologically more plausible. Mathematicians express preferences between theories for similar reasons: e.g. Tarski (1959) prefers a theory (\mathcal{E}_2) over alternatives since '[the conception] embodied in \mathcal{E}_2 distinguishes itself by the simplicity and clarity of its metamathematical implications.'

Our example shows in a detailed, algorithmically oriented manner how to form abstract conceptual domains out of more concrete ones. We make no claims about the cognitive adequacy of our particular choice of these theories.⁶

4.1. Basic domains

The MEASURING STICK domain (MS) refers to acts of constructing linear segments by repeatedly using a linear measuring object. In the version of MS by Lakoff and Núñez (2000), these acts involve using physical segments, which embody distances in real objects such as a wooden stick and which can be used to construct other physical segments, for example when drawing on the ground the limits of a house to be built. The objects of interest are linear segments (straight distances). These are the objects of our formalisation of this domain, cf. Fig. 4.

The second domain of interest is MOTION ALONG A PATH (MAP), in particular motion along a straight path. The basic grounding metaphor maps points on the paths to numbers in arithmetic, just as distances (segments) are mapped to numbers in the MS case. Accordingly, the main objects in our formalisation of the MOTION ALONG A PATH domain are points. MAP is the only one of the four basic grounding domains that includes a natural correspondent to the number zero, namely the ORIGIN. In the formalisation, cf. Fig. 5, the interpretation of *farther*(P_1, P_2) is that point P_1 is farther from the origin than P_2 . Following Lakoff and Núñez, in the basic version of MAP, motion only extends in one direction from the ORIGIN but not in the opposite direction, see law π_4 . The interpretation of *moveAway*(P_1, P_2, P_3) is that if you are at point P_1 on the path and you move away from the origin 'cloning' the path

from the origin to P_2 , then you end up at point P_3 . The interpretations for *closer* and *moveCloser* are similar (see Fig. 5).

The third domain is Lakoff and Núñez's OBJECT CONSTRUCTION domain, including a simplified version of the fraction extension that only contains an operation of splitting objects into halves (see Fig. 6). This will show our point fully while allowing a simple formalisation. The physical setting in mind is one, where objects can be constructed from simpler ones and where some of the objects are classified as WHOLE OBJECTS, meaning that their constituent atomic objects are of a standard size. One can think, for example, of coloured paper-strip constructions made by cutting pieces of paper or sticking them together. Objects that consist of one of several complete sheets of paper stuck together are considered to be WHOLE OBJECTS. A special case of cutting is singled out, which consists of the act of dividing one (constructed) strip of paper into two halves.

In the formalisation, *wholeOb*(O) means that O can be constructed by sticking together whole objects only and *combine*(O_1, O_2, O_3) indicates that the constructed object O_3 is formed by combining objects O_1 and O_2 . Two objects match when they can be exactly superposed; that is, they are of equal size. The meaning of *half*(O_1, O_2) is simply that O_1 is one half of O_2 . The vertical dots at the end of the list of laws indicate that we have laws similar to ϕ_{8a} and ϕ_{8b} for *combine*, *wholeOb*, etc.

It is important to note that, in OCF, the SMALLEST WHOLE OBJECTS play a role similar to that of the UNIT SEGMENT in the MS domain, but while in MS there is a *unique* such object (the particular one used as a ruler), there are many SMALLEST WHOLE OBJECTS in OCF.

An extended version of OCF, one where all the simple fractions of the form $1/n$ are included (at least up to a certain fixed n) can be obtained by presupposing the ability to count up to n . Although we will not work with this extension (let's call it OCF^+), it may look like this.⁷ First, we need an extra sort *num* that stands for \mathbb{N}^+ (or an initial segment of it), the initial numerals 1, 2, 3, etc. as entities of sort *num*, a function symbol *next* that yields the next natural number in the counting sequence and, instead of the predicate symbol *half*, we use two ternary predicates *frac* and *mult*. The intuition is to use *frac*(O_1, n, O_2) to formalise the idea that O_1 is a $1/n$ th of O_2 , while *mult*(O_1, n, O_2) says that constructing an object out of n objects that match O_1 yields an object that matches O_2 . The laws concerning *half* in our OCF are replaced by five laws, the first three ones being similar to the ϕ_6 laws:

$$\begin{aligned} & \forall(O_1, O_2 : obj), (n : nat) : frac(O_1, n, O_2) \leftrightarrow mult(O_2, n, O_1) \\ & \forall(O_1 : obj, n : nat) \exists(O_2 : obj) : frac(O_2, n, O_1) \\ & \forall(O_1, O_2 : obj), (n : nat) : (frac(O_2, n, O_1) \\ & \quad \wedge smallestWhOb(O_1) \rightarrow \neg wholeOb(O_2)) \end{aligned}$$

⁶ On the basis of developmental, historical or educational evidence we would expect that a complete story about how we get to our understanding of rational numbers would start from different formalisations of the domains, include a larger number of blending steps, and involve also other operations such as induction.

⁷ In our opinion, both our versions of OCF are already blends. However, in this paper, we do not concern ourselves with the issue of studying how such a blend is obtained.

MEASURING STICK (MS)

Sorts: *seg*

Entities:

unitSeg : *seg*

Predicates:

longer, shorter : *seg* × *seg*

extend, chop : *seg* × *seg* × *seg*

Laws:

μ_1 : $\forall S_1, S_2 : shorter(S_1, S_2) \leftrightarrow longer(S_2, S_1)$

μ_2 : $\forall S_1, S_2, S_3 : extend(S_1, S_2, S_3) \leftrightarrow chop(S_3, S_2, S_1)$

μ_{3a} : $\forall S_1, S_2 : shorter(S_1, S_2) \vee (S_1 = S_2) \vee longer(S_1, S_2)$

μ_{3b} : $\forall S_1, S_2 : shorter(S_1, S_2) \rightarrow \neg shorter(S_2, S_1)$

μ_4 : $\forall S : \neg longer(unitSeg, S)$

μ_5 : $\forall S_1, S_2, S_3 : extend(S_1, S_2, S_3) \rightarrow longer(S_3, S_2) \wedge longer(S_3, S_1)$

Fig. 4. MEASURING STICK domain (MS).

MOTION ALONG A PATH DOMAIN (MAP)

Sorts: *point*

Entities:

origin : *point*

Predicates:

farther, closer : *point* × *point*

moveAway, moveCloser : *point* × *point* × *point*

Laws:

π_1 : $\forall P_1, P_2 : closer(P_1, P_2) \leftrightarrow farther(P_2, P_1)$

π_2 : $\forall P_1, P_2, P_3 : moveAway(P_1, P_2, P_3) \leftrightarrow moveCloser(P_3, P_2, P_1)$

π_{3a} : $\forall P_1, P_2 : closer(P_1, P_2) \vee (P_1 = P_2) \vee farther(P_1, P_2)$

π_{3b} : $\forall P_1, P_2 : closer(P_1, P_2) \rightarrow \neg closer(P_2, P_1)$

π_4 : $\forall P : \neg farther(origin, P)$

π_{5a} : $\forall P_1, P_2, P_3 : (moveAway(P_1, P_2, P_3) \wedge P_2 \neq origin) \rightarrow farther(P_3, P_2) \wedge farther(P_3, P_1)$

π_{5b} : $\forall P_1, P_3 : moveAway(P_1, origin, P_3) \rightarrow P_1 = P_3$

Fig. 5. MOTION ALONG A PATH domain (MAP).

The other two laws spell out the meaning of *mult*:

$\forall(O_1 : obj) : mult(O_1, 1, O_1)$

$\forall(O_1, O_2, O_3 : obj)(n : nat)$

$(mult(O_1, n, O_2) \wedge combine(O_2, O_1, O_3))$

$\leftrightarrow mult(O_1, next(n), O_3)$

To this we need to add the facts about the initial sequence of natural numbers: $next(1) = 2$, $next(2) = 3$, ... The point is that, conceptually, the ideas we will use in our examples with OCF scale up to OCF⁺. From a computational perspective, however, the set of OCF⁺ laws is harder to work with.

4.2. Conceptual blend 1 – Steps on a path

The original intuition behind the MAP domain (as motivated by Lakoff and Núñez) is that of an entity moving along a straight path. There is no element involving dis-

creteness in such a scenario. By blending the MS and MAP domains we obtain a conception of natural numbers similar to the one used in math classes, in which they are viewed as particular locations on a line. In this section we describe the process leading towards the construction of the conceptual blend, which is partially supported by the HDTP framework.

4.2.1. Exploration

HDTP finds analogical mappings between the two domains, as a byproduct of calculating a generalising domain *G*. See Fig. 7 for a depiction of the result produced by HDTP when we take MS as the source.

In the general case, several different sets of analogical mappings might be found. (This also depends on which one of the domains we use as target and which one as source.) In the particular case of MS and MAP, HDTP will

OBJECT CONSTRUCTION WITH FRACTIONS (OCF)

Sorts: obj

Predicates:

<i>smallestWhOb, wholeOb</i>	: obj
<i>larger, smaller</i>	: obj × obj
<i>matches</i>	: obj × obj
<i>combine, split</i>	: obj × obj × obj
<i>half</i>	: obj × obj

Laws:

ϕ_1	: $\forall O_1, O_2 : \text{smaller}(O_1, O_2) \leftrightarrow \text{larger}(O_2, O_1)$
ϕ_2	: $\forall O_1, O_2, O_3 : \text{combine}(O_1, O_2, O_3) \leftrightarrow \text{split}(O_3, O_2, O_1)$
ϕ_{3a}	: $\forall O_1, O_2 : \text{smaller}(O_1, O_2) \vee \text{matches}(O_1, O_2) \vee \text{larger}(O_1, O_2)$
ϕ_{3b}	: $\forall O_1, O_2 : \text{smaller}(O_1, O_2) \rightarrow \neg \text{smaller}(O_2, O_1)$
ϕ_4	: $\forall O_1, O_2 : \text{smallestWhOb}(O_1) \wedge \text{wholeOb}(O_2) \rightarrow \neg \text{larger}(O_1, O_2)$
ϕ_5	: $\forall O_1, O_2, O_3 : \text{combine}(O_1, O_2, O_3) \rightarrow \text{larger}(O_3, O_2) \wedge \text{larger}(O_3, O_1)$
ϕ_{6a}	: $\forall O_1, O_2, O_3 : \text{matches}(O_1, O_2) \rightarrow (\text{half}(O_1, O_3) \leftrightarrow \text{combine}(O_1, O_2, O_3))$
ϕ_{6b}	: $\forall O_1 \exists O_2 : \text{half}(O_2, O_1)$
ϕ_{6c}	: $\forall O_1, O_2 : (\text{half}(O_2, O_1) \wedge \text{smallestWhOb}(O_1) \rightarrow \neg \text{wholeOb}(O_2))$
ϕ_7	: $\forall O : \text{smallestWhOb}(O) \rightarrow \text{wholeOb}(O)$
ϕ_{8a}	: $\forall O_1, O_2 : \text{matches}(O_1, O_2) \rightarrow (\text{smallestWhOb}(O_1) \leftrightarrow \text{smallestWhOb}(O_2))$
ϕ_{8b}	: $\forall O_1, O'_1, O_2, O'_2 : (\text{matches}(O_1, O'_1) \wedge \text{matches}(O_2, O'_2) \rightarrow (\text{smaller}(O_1, O_2) \leftrightarrow \text{smaller}(O'_1, O'_2))$
	: ...

Fig. 6. OBJECT CONSTRUCTION with fractions (OCF).

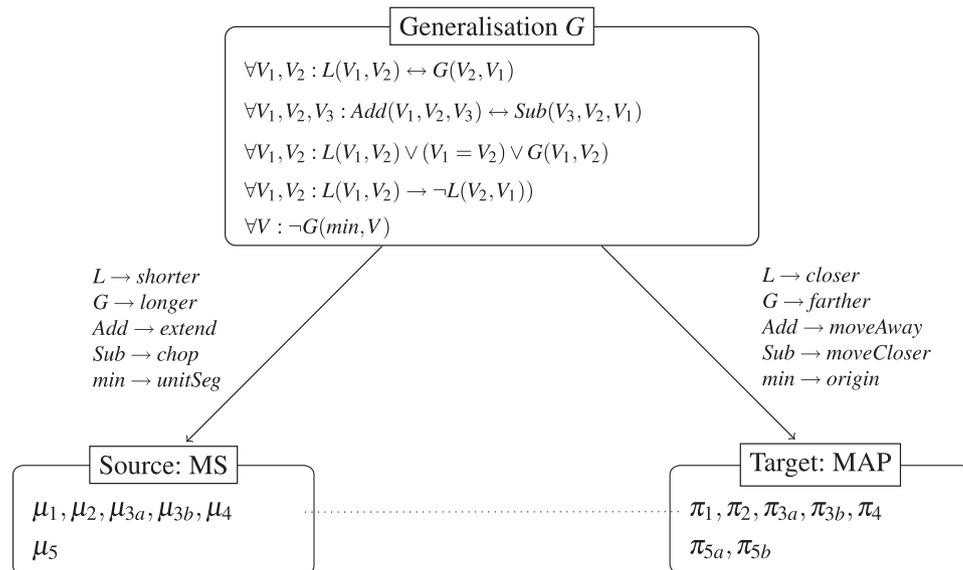


Fig. 7. Generalisation produced by HDTP from the MS and MAP domains.

find that laws μ_1 to μ_4 of MS correspond to π_1 – π_4 of MAP, regardless of which of the two domains is taken as the source.

4.2.2. Goal formulation and discoveries

The most immediate goal after establishing an analogical mapping is to try to transfer properties from the source domain to the target. However, in our example, μ_5 cannot be transferred to MAP because we can derive a contradiction from the formula obtained from μ_5 by renaming

according to the alignments and π_{5b} . Going in the other direction, we could try to transfer the π_5 laws to MS. In the case of π_{5a} this is possible, but the transferred formula is already implied by μ_5 , so the transfer does not yield anything new. On the other hand, π_{5b} cannot be transferred because this also creates a contradiction.

A second way to take advantage of the analogical mapping is to try to create a conceptual blend. Intuitively, such a blend may be based on the observation that there is a good structural match between the domains, but that transfers fail

because a certain object exists in MAP that does not exist in MS. This may trigger the conjecture that MAP could be a more general domain than MS. Thus, a goal can be formulated to create a conceptual blend in which MS is embedded in MAP. In such a blend, the objects would have a dual nature and could be thought of as distances or points, and there would be a subdomain of the global one that would have the discrete nature of the MS objects. Such a conceptual blend does exist. Once we have made the conjecture that the notion of points on a line is more general than the notion of distances built out of a unit measure, one can construct the blend as follows, with an eye on detecting possible inconsistencies as new laws are added to it.

4.2.2.1. Sorts. The conceptual blend uses only one sort, the same one that is used in the domain that seems to be ‘larger’ (in this case MAP).

4.2.2.2. Signature. The signature of the conceptual blend is given by the ‘union’ of the signatures of MS and MAP, which is created simply by putting both signatures together and changing the occurrences of *seg* in the signature definitions to *point*. Furthermore, we add a new monadic predicate symbol *nat*. The intuition is that $nat(P)$ corresponds (via the blending mappings) to a point constructed in an MS-like way. At an even more intuitive level, the objects P satisfying $nat(P)$ would be points that can be constructed by iterating steps of a unit size. These points are like the locations of natural numbers on the real line. Note that the introduction of *nat* can be seen as a special case of predicate introduction in blend signatures as described in Section 4.3.

4.2.2.3. Laws. Here we include:

1. All laws of MAP, as they should hold in the larger universe.
2. All alignments between predicate symbols found by HDTP, in the form of equivalences.
3. For each entity symbol in the ‘smaller’ domain (MS in this case), a law stating that each such entity belongs to the subdomain of the conceptual blend. That is, we add $nat(unitSeg)$ to the theory of the blend.
4. For each law μ_i of MS, add a relativised version μ'_i of it to the blend. For example, the relativised versions of μ_1 , μ_4 , and μ_5 are:

$$\mu'_1 : \forall S_1, S_2 : (nat(S_1) \wedge nat(S_2)) \\ \rightarrow (shorter(S_1, S_2) \leftrightarrow longer(S_2, S_1))$$

$$\mu'_4 : \forall S : nat(S) \rightarrow \neg longer(unitSeg, S)$$

$$\mu'_5 : \forall S_1, S_2, S_3 : (nat(S_1) \wedge nat(S_2) \wedge nat(S_3)) \\ \wedge extend(S_1, S_2, S_3) \rightarrow longer(S_3, S_2) \wedge longer(S_3, S_1))$$

4.2.2.4. Refine the laws. Several of the relativised laws can actually be dropped from the theory or changed to unrelativised versions. For example, it is clear that μ'_1 is simply

a special case of π_1 , so the law can be changed to μ_1 (unrelativised) or dropped. On the other hand, μ'_4 is an important addition to the blend that cannot be derived from the other laws added so far.

We think of the resulting blend, depicted in Fig. 8, as one whose resulting domain is that of DISCRETE MOTION ALONG A PATH (DMAP; discrete meaning *stepped* here).

A more concise version of the blend can be given by mapping correlated predicates to an *identical* predicate in the blend (here, for example, *longer* and *farther*). The choice of terminology does not matter in Goguen’s formulation of blending, as long as the mappings between conceptualisations match up correctly; we expect cognitive factors to play a role here however, and so permit both inherited predicates into the blend.

In this example, the process ends felicitously, because no inconsistencies appeared as new laws were added while forming the blend. If at some point an inconsistency would have been detected, the conjecture that initially triggered the blend formation process would not have been supported. Nevertheless, at the step previous to the inclusion of a problematic law we would still have a blend, even if not the expected one.

4.2.3. Testing and discovery

Once the theory of the blend domain is assembled, one can try to improve it or use it to make discoveries (which may involve emergent structure in the blend). A first strategy for exploration in this example is the question of whether the symbols of constants that were collapsed in the generalisation domain can also be collapsed in the conceptual blend. However, adding $unitSeg = origin$ to DMAP creates a contradiction. Looking at the arithmetical counterpart of this, the conceptual blend entails a version of the arithmetical fact that $0 \neq 1$. This finding leads to a revision of the generalisation and substitution functions, so that the identification between *unitSeg* and *origin* is dropped (Fig. 8 does not include this revision).

In this conceptual blend it is also a fact that there are points strictly between 0 and 1, which one can still think of and talk about as segments or distances (from the origin to the point), just that their exact distance from the origin is not measurable by using unit steps and the available operations.

4.3. Searching for conceptual blends

The ideas used in the previous example can be generalised. The way to do this leads to an explosion of possibilities that could be tried, and, consequently, to the need to find good heuristics to choose which possibilities to try first. For example, we can start with two domains described by theories Th_1 , Th_2 , which are given in signatures Σ_1 , Σ_2 , respectively. Suppose that an analogical mapping has been found between both domains, with the property that for all function symbols, if f_1 is mapped to f_2 , then f_1 and f_2 have the same arity, and suppose that a similar property holds for analogical

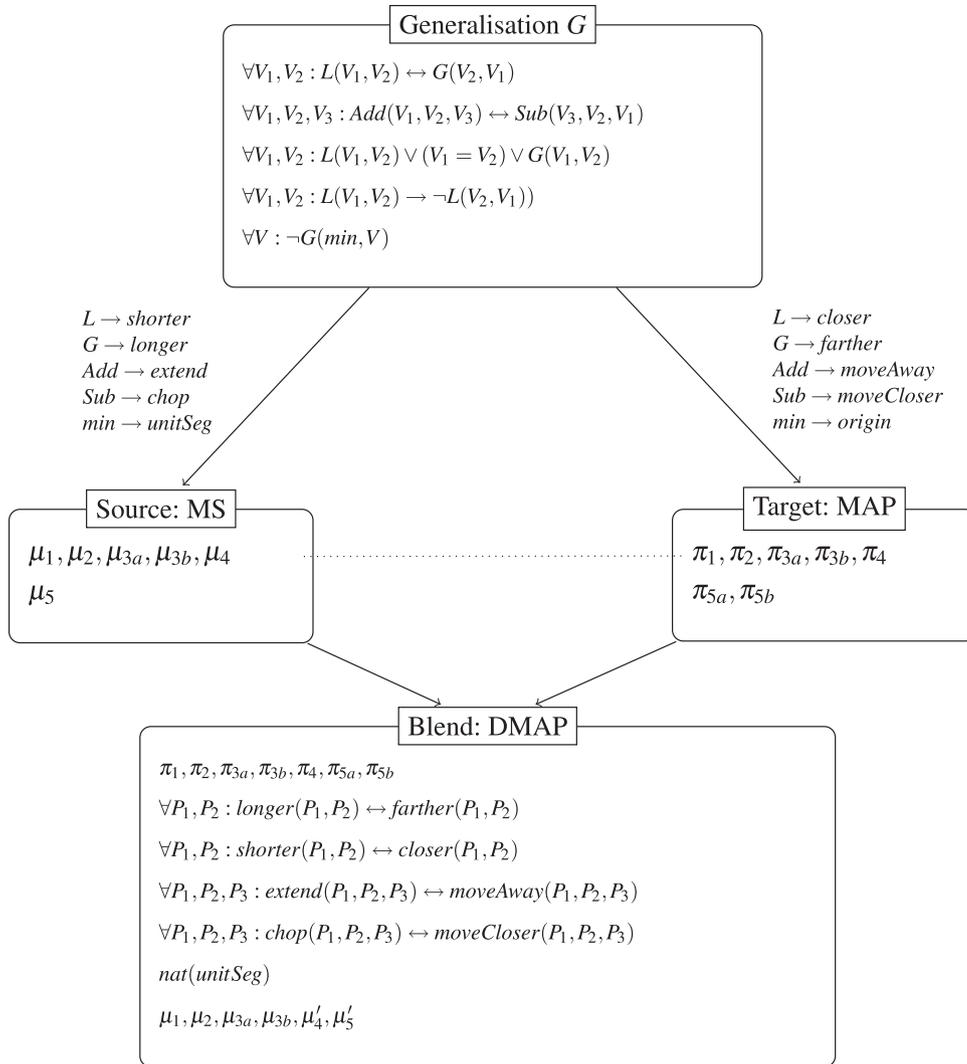


Fig. 8. First conceptual blend on the basis of the generalisation shown in Fig. 7.

mappings between relation symbols. This is a way to say that an analogical mapping was found, where the substitutions are of the simplest kind (renaming and fixations). Then one can attempt to create blend Th_3 as follows:

4.3.1. Sorts

Independently of the number of sorts in the input domains, we create a conceptual blend with only one sort, which we call *univ*. Of course, for all intents and purposes this is the same as using a non-sorted language. The reason for this choice will become clear when we explain how to construct the next components of the theory blend.

4.3.2. Signature

The signature of the conceptual blend is given by the ‘union’ of the signatures of MS and MAP, which is now achieved by putting both signatures together and changing all sorts in the signature definitions to *univ*. To this, we add a new monadic predicate symbol *sort* for each sort used in $\Sigma_1 \cup \Sigma_2$ (without loss of generality we assume that $\Sigma_1 \cap \Sigma_2 = \emptyset$). What we are doing here is moving the sorts

of the original domains to the language of the conceptual blend.

4.3.3. Core blend laws

1. For each formula in theories Th_1 and Th_2 that is covered by the analogy, add its relativised form to the blend. That is, for each such formula we recursively change formulas $\forall x: \mathbf{Sort}(\phi)$ to $\forall x(sort(x) \rightarrow \phi)$. Similarly, we change $\exists x: \mathbf{Sort}(\phi)$ to $\exists x(sort(x) \wedge \phi)$. Then we add the result to the conceptual blend.
2. For each entity symbol e of sort \mathbf{Sort} in any of the input domains, if the symbol was covered by the analogy, then add a fact $sort(e)$ to the blend.
3. For each function symbol $f: \mathbf{Sort}_1 \times \dots \times \mathbf{Sort}_n \rightarrow \mathbf{Sort}$ in any of the input domains, if the symbol was covered by the analogy, then add the following formula to the conceptual blend:

$$\forall x_1, \dots, x_n ((sort_1(x_1) \wedge \dots \wedge sort_n(x_n)) \rightarrow sort(f(x_1, \dots, x_n)))$$

4.3.4. Preferred conjectures

In addition to the previous core of laws, the following ones can be added, testing at each step for possible inconsistencies of the resulting theory:

1. If two entities e_1 and e_2 have been identified by the analogy, add $e_1 = e_2$ to the conceptual blend.
2. If the function f has been identified via the analogy with the function g , then add

$$\forall x_1, \dots, x_n (f(x_1, \dots, x_n) = g(x_1, \dots, x_n)).$$
3. If the relation R has been identified via the analogy with the relation Q , then add

$$\forall x_1, \dots, x_n (R(x_1, \dots, x_n) \leftrightarrow Q(x_1, \dots, x_n)).$$
4. Add a theory about desirable containment or overlapping relations about sorts (including the universal one), that is, add a number of statements of the form $\forall x (sort_1(x) \rightarrow sort_2(x))$, $\exists x (sort_1(x) \wedge sort_2(x))$ or $\forall x (sort_1(x))$.

4.3.5. Extra conjectures

A consistent conceptual blend found in the previous step can be enriched with core blend laws and preferred conjectures in any following ways:

1. If a law that involves only symbols covered by the analogy is not part of the generalisation, try to add it (in relativised form) to the conceptual blend.
2. If a symbol in the signature in one of the input domains is not covered by the analogy, try to add it to the conceptual blend (plus at least one of the laws related to the symbol, in relativised form).

Nothing prevents a formulation of the previous process that works directly with multi-sorted languages so that the resulting blend would be a theory in such a language. No *sort* predicates would be introduced, and our *preferred conjecture 4* would turn into a formal inquiry about the hierarchy of sorts. In our example, *nat* would be introduced as a new sort. The implementation of HDTP's capabilities at reasoning with first order logic enriched with hierarchies of sorts is on the way but at an initial stage yet.

It is clear that, in general, given an analogical mapping, the number of conjectured blends may be huge. This means that good heuristics are of prime importance. Our example with MS and MAP makes use of one such heuristic: conjecture that one of the domains is a subdomain of another one if both domains seem identical, except if one of them has elements of a new type (satisfying formulas that no element of the other domain can satisfy). Concretely, in the example, the conjecture is made that there is a blend in which the objects x satisfying *nat*(x) also satisfy *point*(x), and that *point* corresponds to the universal sort. This last part of the conjecture allows us to use a simpler notation in the example, by using *point* as the main sort rather than as a monadic predicate in the signature.

In general, for a simple example like ours, where we have two domains, each with only one sort, the blend domain will include two predicates representing sorts. In total, there are ten possible overlapping configurations for these sorts: the two imported sorts may be disjoint, may be the same, one may be strictly contained in the other (two possibilities here), or they may have a nonempty intersection without one sort being contained in the other. For each one of these five possibilities it may be that the union of the sorts is the whole universe of the blend, or it may be

DISCRETE MOTION ALONG A PATH DOMAIN (DMAP)

Sorts: point

Entities:

origin, unitSeg : point

Predicates:

nat : point
farther, closer, longer, shorter : point \times point
moveAway, moveCloser, extend, chop : point \times point \times point

Laws:

$\pi_1, \pi_2, \pi_{3a}, \pi_{3b}, \pi_4, \pi_{5a}, \pi_{5b}$

$\mu_1, \mu_2, \mu_{3a}, \mu_{3b}$

μ'_4 : $\forall S : nat(S) \rightarrow \neg longer(unitSeg, S)$
 μ'_5 : $\forall S_1, S_2, S_3 : (nat(S_1) \wedge nat(S_2) \wedge nat(S_3) \wedge extend(S_1, S_2, S_3) \rightarrow longer(S_3, S_2) \wedge longer(S_3, S_1))$
 β_1 : $\forall P_1, P_2 : longer(P_1, P_2) \leftrightarrow farther(P_1, P_2)$
 β_2 : $\forall P_1, P_2 : shorter(P_1, P_2) \leftrightarrow closer(P_1, P_2)$
 β_3 : $\forall P_1, P_2, P_3 : extend(P_1, P_2, P_3) \leftrightarrow moveAway(P_1, P_2, P_3)$
 β_4 : $\forall P_1, P_2, P_3 : chop(P_1, P_2, P_3) \leftrightarrow moveCloser(P_1, P_2, P_3)$
 β_5 : $nat(unitSeg)$

Fig. 9. Specification of the DISCRETE MOTION ALONG A PATH domain.

that it is not. A theory about these containment relations may be one that actually does not determine exactly one of the ten possibilities.

4.4. Conceptual blend 2 – a subset of the rational numbers

Coming back to our example, we may now try to get a conceptual blend of DMAP and OCF. In fact, we are interested in how to get to a blend that corresponds to the idea of rational numbers (in our case a subset of them) as points on the line. As always, the first step is to find an analogical mapping between the two domains. The DMAP domain is fully spelled out in Fig. 9.

According to the HDTP algorithm for analogy making, taking OCF as source and DMAP as target, it is possible to find the following alignment between the signature symbols of both domains:

OCF	DMAP
smallestWhOb	$\lambda x.(x = \text{unitSeg})$
wholeOb	nat
larger	longer
smaller	shorter
combine	extend
split	chop
matches	=

Of special interest in this table is the first row, which actually stands for an alignment between the monadic relation *smallestWhOb* in OCF and the monadic relation $x = \text{unitSeg}$ in DMAP.

Conceptually, the overall effect of this alignment is that all the smallest whole objects in OCF are at once identified with the unit object of DMAP.

At the implementation level, HDTP goes through some key steps. First, there is a ‘normalisation’ process by which the axiom μ'_4 of DMAP is rewritten to the equivalent

$$\forall S : (x = \text{unitSeg} \wedge \text{nat}(S)) \rightarrow \neg \text{longer}(x, S).$$

This re-represented axiom matches in structure with ϕ_4 from OCF, which allows HDTP to notice that $x = \text{unitSeg}$ and *smallestWhOb*(x) play similar roles. The generalisation theory obtained by HDTP consists of formulas each one of which generalises one of the following pairs of laws:

OCF	DMAP
ϕ_1	μ_1
ϕ_2	μ_2
ϕ_{3a}	μ_{3a}
ϕ_{3b}	μ_{3b}
ϕ_4	μ''_4

where μ''_4 is the re-representation of μ_4 described above⁸.

The alignment between the binary predicate *matches* in OCF and true equality in DMAP (also reflected by the correspondence between the laws ϕ_4 and μ'_4) suggests a heuristic for creating a blend that essentially is the mathematical idea of exploring whether *matches* is a congruence with respect to the operations of the domain. If this is the case, then a quotient-like construction can be performed with the effect that operations in the source domain for which *matches* works as a congruence may be tried to be imported into the target domain.

A conceptual blend in this case may be created as follows:

1. The conjecture is made that *matches* works really as an equality. To start testing this, add to OCF the laws saying that *matches* is an equivalence relation. If the theory is consistent then call this theory Th_1 .
2. For each n -ary relation symbol R in OFC that is covered by the analogical mapping, test to see if Th_1 together with

$$\forall x_1, y_1, x_2, y_2, \dots, x_n, y_n ((\text{matches}(x_1, y_1) \wedge \dots \wedge \text{matches}(x_n, y_n)) \rightarrow (R(x_1, \dots, x_n) \rightarrow R(y_1, \dots, y_n)))$$

is still consistent. If it is, then add to the blend the natural equivalence induced by the analogical mapping between R and its counterpart in DMAP. For instance, in our example, the law ϕ_{8b} plus the fact that *smaller* was analogically mapped to *shorter* says that we can add the law

$$\forall x, y (\text{smaller}(x, y) \leftrightarrow \text{shorter}(x, y))$$

to the conceptual blend. Similarly, if it is consistent then we can include in the blend the equivalences for all the alignments of the first table. In particular, the natural equivalence that corresponds to the first such alignment is

$$\forall x (\text{smallestWhOb}(x) \leftrightarrow x = \text{unitSeg}).$$

3. Add to the blend all the DMAP versions of all the laws in the second table (the laws covered by the analogy) that include only predicates that are part of the signature of the blend introduced in the last step.
4. Try to import more predicates and laws from DMAP, while maintaining the goal to preserve consistency. Similarly, try to import laws from OCF, writing ‘=’ instead of *matches* in the case, where *matches* behaves as a congruence with respect to any new signature symbol involved. In particular, assuming that the laws of OFC include (or are enough to deduce) that

⁸ The axiom ϕ_4 includes the subformula *smallestWhOb*(x), while μ'_4 includes the subformula $x = \text{unitSeg}$ (something of the form $P(x, c)$, where P is equality and the constant c is *unitSeg*). One of the higher-order anti-unification legal forms of substitution allows to anti-unify $P(x, c)$ and *smallestWhOb*(x) into something of the form $Q(x)$, which is what appears in the generalisation theory that is created together with the alignments.

$$\forall O_1, O'_1, O_2, O'_2 ((\text{matches}(O_1, O'_1) \wedge \text{matches}(O_2, O'_2)) \\ \rightarrow (\text{half}(O_1, O_2) \leftrightarrow \text{half}(O'_1, O'_2))),$$

try to include *half* in the signature of the blend, together with laws ϕ_6 .

At the level of its theory, the resulting conceptual blend in this case turns out to be an expansion of DMAP with the equivalences induced by the alignments of the first table above, and all the laws of OCF that did not have an analogical counterpart according to the second table. All in all, this example can also be seen as a case of a rich analogical transfer from DMAP to OCF. However, the strategy described here can be generalised straightforwardly to a general setting, where not necessarily all laws of the source and input domains may be imported to the conceptual blend.

In the process of construction of this second conceptual blend we can (see before) distinguish a stage of exploration (given by the finding of analogical mappings), the formulation of a goal (here based on the conjecture that *matches* is some kind of equality), the working towards fulfilling the goal, and a last stage of discovery in the conceptual blend. In particular, it is true in this blend (in the domain of paths) that certain distances *exist*, namely all those whose construction can be described by a quantifier free, closed formula in the theory of the blend.⁹ Those points are precisely the ones corresponding to the rational numbers of the form $n/2^k$ for some $n, k \in \mathbb{N}$.

4.5. Some comments

We have given an account, supported by a partial implementation, of how different underlying concepts of number may be blended. Some comments are needed on how this HDTP presentation relates to Goguen's characterisation, and also on the comparison with re-representation ideas from the structure mapping community.

Our presentation is an instantiation of Goguen's account of conceptual blending in the following way. The computed generic spaces are mapped to input spaces via substitutions, which correspond to signature morphisms in Goguen's approach (here, working in the category of sorted first-order logic) with slight extensions, where re-representation in the sense of Yan, Forbus, and Gentner (2003) allows logically equivalent forms to be employed to reveal the structural similarity. At any event, these maps preserve inference patterns from one space to another.

The blend spaces in HDTP introduce explicit equivalences between syntax that has been aligned via the generic space. Goguen's approach, where the diagram commutes,

enforces a choice between the syntax of the given spaces; the approach here is simply a definitional extension of Goguen's characterisation, where the syntaxes are interchangeable. Cognitively, there may be reason to prefer one syntax or indeed invent some new term.

For given generic spaces and mappings to input spaces, Goguen suggests that the conceptual blend should be a push-out in category theoretic terms. This is the most general space that can be obtained from combining the given spaces, while respecting the shared generic properties. (This is a version of the 'maximal structurally consistent' combination of Yan et al. (2003)).

At various points, the description above calls for assurances that some posited space is logically consistent. We are aware that this is undecidable in general, so what does this assurance call for? On the one hand, in simple cases, proofs of inconsistency can be found relatively easily. On the other hand, another aspect of our approach which we do not have space to elaborate on here, involves not only considering syntactic structure, but also relating operations over syntax to semantic characterisations of the situations described by these theories. In the case at hand, we can model the OBJECT CONSTRUCTION space (for example) via appropriate data-types and effective functions. Then an assurance of consistency may come from the existence of a model, in these terms, for a given conjectured theory. For general accounts of this semantic counterpart to the work described above see Goguen (2006) and Krumnack, Gust, Schwering, and Kühnberger (2010).

Finally, we have described methods for constructing conceptual blends and heuristics for constraining the search for these blends. There is a family resemblance with the methods and heuristics of Yan et al. (2003). The blending operations here give a principled way to investigate situations, where a variety of blend spaces can be brought into play alongside and related to the initial spaces.

5. Conclusions

Although there already exist a number of computational systems for discovering mathematical concepts, none of these systems takes into account cognitive aspects involved in mathematical discoveries. We started from the basic conceptions of Lakoff and Núñez (2000), who lay out how elementary arithmetic is grounded in concrete, physical domains, and of Fauconnier and Turner (2002), who describe a general framework of conceptual blending. Both conceptions are presented in a non-computational manner, and Fauconnier and Turner (2008, p. 59) even state explicitly that 'blending is not algorithmic'. We, however, think that the overall framework of conceptual blending is indeed suited for a computational account. This account is similar to Goguen's (2006) approach, but we used the analogy engine HDTP as our point of departure.

We used HDTP to compute a generalisation of a source and a target domain and extended HDTP in such a way that a conceptual blend can be computed based on the ana-

⁹ When we talk here about existence of the points or distances, we have in mind an idea of existence of numbers close (but not identical) to that used by early Greeks (e.g. Pythagoreans), who required the existence of a construction based on certain geometrical rules in order to accept a distance as a number.

logical relation between a source and a target domain that goes beyond simple structural alignment. The examples that we discussed are three of the four basic metaphors for arithmetic by Lakoff and Núñez (2000) that ground elementary arithmetic in concrete, physical domains: OBJECT CONSTRUCTION, MEASURING STICK and MOTION ALONG A PATH.

A main point of this paper is that conceptual blending is an important means for making new discoveries in mathematics, cf. also Alexander (2009). Creating analogies and making analogical transfers is insufficient for this purpose. A motivating, non-mathematical case is the conceptual blending of the concepts HOUSE and BOAT into HOUSEBOAT and BOATHOUSE, where it is highly implausible to abandon the original conceptual spaces for HOUSE and BOAT once the more complex concepts have been created via blending. And for mathematics, Fauconnier and Turner (2002) make this point for blending notions of number which are discrete in one case and dense in another. In the case we discussed in this paper, it seems quite sensible that notions of equality (equivalence) of fractions should exist side by side that say that $1/2$ and $2/4$ are different (because they are constructed from different numbers) as well as the same (because they correspond to the same rational number). This, however, can only be done if the two notions exist in different conceptual spaces, because, otherwise, the representation would be inconsistent.

Future work will now include continuing to implement the conception we laid out in this paper, which in particular will involve applying our approach to a larger number of case studies and identifying which of the mentioned heuristics will be the most fruitful for mathematical discovery.

Acknowledgments

We would like to thank the two anonymous reviewers for the comments and suggestions we received.

Markus Guhe, Alison Pease and Alan Smaill were supported by EPSRC Grant EP/F035594/1 for the *Wheelbarrow* project.

References

- Alexander, J. (2009). *Mathematical blending*. Working Paper. Case Western Reserve University.
- Arnauld, A., & Nicole, P. (1683). *La logique, ou l'art de penser*. Paris: G. Desprez.
- Arzarello, F., Andriano, V., Olivero, F., & Robutti, O. (1998). Abduction and conjecturing in mathematics. *Philosophica*, 61.
- Colton, S. (2002). *Automated theory formation in pure mathematics*. Springer-Verlag.
- Corfield, D. (2003). *Towards a philosophy of real mathematics*. Cambridge: Cambridge University Press.
- Epstein, S. L. (1988). Learning and discovery: One system's search for mathematical knowledge. *Computational Intelligence*, 4, 42–53.
- Fajtlowicz, S. (1988). On conjectures of graffiti. *Discrete Mathematics*, 72, 113–118.
- Fauconnier, G., & Turner, M. (2002). *The way we think: Conceptual blending and the mind's hidden complexities*. New York: Basic Books.
- Fauconnier, G., & Turner, M. (2008). Rethinking metaphor. In R. Gibbs (Ed.), *Cambridge handbook of metaphor and thought* (pp. 53–66). New York: Cambridge University Press.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155–170.
- Gentner, D., Bowdle, B. F., Wolff, P., & Boronat, C. (2001). Metaphor is like analogy. In D. Gentner, K. J. Holyoak, & B. N. Kokinov (Eds.), *The analogical mind: Perspectives from cognitive science* (pp. 199–253). Cambridge, MA: MIT Press.
- Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American Psychologist*, 52, 45–56.
- Goguen, J. (2006). Mathematical models of cognitive space and time. In D. Andler, Y. Ogawa, M. Okada, & S. Watanabe (Eds.), *Reasoning and cognition: Proceedings of the interdisciplinary conference on reasoning and cognition* (pp. 125–128). Keio University Press.
- Gray, J. (1992). The nineteenth century revolution in mathematical ontology. In D. Gillies (Ed.), *Revolutions in mathematics* (pp. 226–248). Oxford: Clarendon Press.
- Guhe, M., Pease, A., Smaill, A., Schmidt, M., Gust, H., Kühnberger, K.-U., et al. (2010). Mathematical reasoning with higher-order anti-unification. In *Proceedings of the 32nd annual conference of the Cognitive Science Society*.
- Harris, J. (1992). Developments in algebraic geometry. In *Proceedings of the AMC centennial symposium*. Providence: A.M.S. Publications.
- Hofstadter, D. R., & Mitchell, M. (1994). The copycat project: A model of mental fluidity and analogy-making. In K. Holyoak & J. Barnden (Eds.), *Analogical connections. Advances in connectionist and neural computation theory* (Vol. 2, pp. 31–112). Ablex Publishing Corporation.
- Kaplan, R. (2000). *The nothing that is: A natural history of zero*. USA: Oxford University Press.
- Keane, M. (1990). Incremental analogizing: Theory and model. In K. J. Gilhooly, M. G. T. Keane, R. H. Logie, & G. E. Erds (Eds.), *Lines of thinking: Reflections on the psychology of thought: Representation, reasoning, analogy and decision making* (pp. 221–235). Chichester, UK: John Wiley and Sons.
- Krumnack, U., Gust, H., Schwering, A., & Kühnberger, K.-U. (2010). Remarks on the meaning of analogical relations. In M. Hutter, E. Baum, & E. Kitzelmann (Eds.), *Artificial General Intelligence, 3rd international conference AGI*. Atlantis Press.
- Krumnack, U., Schwering, A., Gust, H., & Kühnberger, K.-U. (2007). Restricted higher-order anti-unification for analogy making. In *AI 2007: Advances in Artificial Intelligence. Lecture notes in artificial intelligence* (Vol. 4830, pp. 273–282). Berlin: Springer.
- Lakatos, I. (1976). *Proofs and refutations*. Cambridge, UK: Cambridge University Press.
- Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago: The University of Chicago Press.
- Lakoff, G., & Núñez, R. E. (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being*. New York: Basic Books.
- Lenat, D. (1976). *AM: An artificial intelligence approach to discovery in mathematics*. Ph.D. thesis, Stanford University.
- Mancosu, P. (2008). *The philosophy of mathematical practice*. Oxford: Oxford University Press.
- Nersessian, N. J. (2008). *Creating scientific concepts*. Cambridge, MA: MIT Press.
- Newell, A. (1990). *Unified theories of cognition: The William James lectures, 1987*. Cambridge, MA: Harvard University Press.
- Plotkin, G. D. (1970). A note on inductive generalization. *Machine Intelligence*, 5, 153–163.
- Polya, G. (1954). *Mathematics and plausible reasoning. Induction and analogy in mathematics* (Vol. 1). Princeton University Press.
- Polya, G. (1962). *Mathematical discovery*. New York: John Wiley and Sons.
- Schwering, A., Krumnack, U., Kühnberger, K.-U., & Gust, H. (2009). Syntactic principles of heuristic-driven theory projection. *Journal of Cognitive Systems Research*, 10, 251–269.

- Sims, M., & Bresina, J. (1989). Discovering mathematical operator definitions. In *Proceedings of the sixth international workshop on machine learning*. S.F., CA: Morgan Kaufmann.
- Stewart, I. (2007). *Why beauty is truth: The history of symmetry*. NY, USA: Basic Books.
- Tappenden, J. (2008). Mathematical concepts and definitions. In P. Mancosu (Ed.), *The philosophy of mathematical practice* (pp. 256–275). Oxford: Oxford University Press.
- Tarski, A. (1959). What is elementary geometry? In L. Henkin, P. Suppes, & A. Tarski (Eds.), *The axiomatic method, with special reference to geometry and physics* (pp. 16–29). Amsterdam: North-Holland Publishing Company.
- Yan, J., Forbus, K. D., & Gentner, D. (2003). A theory of representation in analogical mapping. In *Proceedings of the 25th annual meeting of the Cognitive Science Society*. Lawrence Erlbaum.