Incorporating Reinterpretation in Analogical Reasoning

Hannah Deininger

Seminar KI: gestern, heute, morgen
Angewandte Informatik, Universität Bamberg
Winter Term 2017/2018

Abstract. This paper aims to give an overview of the theoretical background regarding Analogical Reasoning in Artificial Intelligence (AI), as well as presenting the option of incorporating reinterpretation in analogical reasoning. The focus lies on the handling of geometric proportional analogies with computer programs. Starting with basic concepts of analogical reasoning, the first approach of implementing analogical reasoning in AI by Evans (1964) is presented. The following part of the paper deals with the idea of integrating reinterpretation in Analogical Reasoning approaches in AI research by O'Hara and Indurkhya (1994). Finally, the paper tries to discuss the strengths and limitations according to the presented practical approaches.

1 Introduction

> It is proper ... to consider the similar, even in things far distant from each other.

As this quote from Aristotle (Metaphysics) states, Analogical Reasoning is fundamental to human thinking and was already important in the beginning of the human culture in the Ancient Greece. Beginning at young age, humans tend to gain a lot of knowledge over their lifespan through analogical reasoning. Also, analogical problem solving plays a major role during our life time. In mathematics classes, for example, knowledge is conveyed by teachers giving examples during lessons and students solving analogical problems with the help of similiar case's solutions. Another context of usage for analogies is explaining content of unknown domains by referencing to known domains.

Ever since the Greeks, analogical reasoning has been recognized as a key phenomenon of intelligence (Kedar-Cabelli, 1988). This shows analogies' great importance for the human cognition. Mostly, analogical reasoning is highly influential as it allows to transfer knowledge from a well-understood situation (the base) to explain a less familiar situation (the target), with tremendous savings of reasoning. Furthermore, understanding of analogies is crucial to understanding metaphors used in natural language (Kedar-Cabelli, 1988).

As analogical reasoning has such importance to humans, computer scientists started to transfer this human ability to computer systems in order to better understand analogical reasoning and problem solving and to get stronger AI systems. Therefore, it is not surprising that the topic analogical reasoning has a long history in AI research (Hall, 1989), mainly because of the opportunity to effectively use knowledge. Based on the mechanisms for connecting the source with the target domain, analogy provides a basic machinery for linking the reasoner's past and present experience (Hall, 1989). Additionally, analogy is very valuable for software engineering. In this context a systematic reuse of well-functioning and efficient code fragments is highly supported, as this can reduce the development costs dramatically.

In this paper an overview of basic theoretical concepts as well as two explicit practical approaches of analogical reasoning shall be given. Hereinafter, the focus shall be on the first approach to analogical reasoning by Evans (1964) and the incorporation of reinterpretation in analogical reasoning by O'Hara and Indurkhya (1994).

2 Basic Concepts of Analogical Reasoning

The following chapter will give an overview of the theoretical concepts regarding Analogical Reasoning. When discussing a topic in a scientific way it is essential to firstly define necessary key terms. For this paper, this will be done in the next paragraph.

2.1 Analogy

In general, an analogy is the comparison of two objects from different domains, which are thought to be similar in some features or relations (DeLoache, Miller, & Pierroutsakos, 1998). The goal is to explain or predict unknown aspects of one domain, called target domain, by exploiting experience from the well-known second domain, called base domain.

In the field of analogy, it is furthermore relevant to mention the degree of silmilarity between the base and the target domain. In general, it is spoken of an analogy if the attributes mapped from the base to the target are few, whereas the mapped relations are many. This makes analogies to a preliminary stage of abstraction. The abstraction over concepts and generalisation over domains is absolutely necessary to achieve analogies (Gentner, 1983).

There are three categories of analogies (Schmid, Gust, Kühnberger, & Burghardt, 2003):

1. *Proportional Analogy:*
   This is the simplest form of analogies: Transfer of just one relation
   Example: Evening is to Day as Old Age is to Life → Transfer of the relation "last part of"
2. *Predictive/Explanatory:*
   Transfer of many known principles to a new unknown domain
   Example Rutherford Analogy:

The Rutherford Analogy is an approach to explain the behaviour of atoms by relating it to the behaviour of the solar system, which was more familiar: The structure of the hydrogen atom is like the structure of the solar system → Transfer of many principles valid in the solar system to the hydrogen atom

3. *in Problem Solving:*
   Within one domain former examples are used and a old problem's known solution is transferred to a new problem
   Example: Mathematical or programming problems → Use of old functioning program code of similar programming tasks to solve new tasks

This paper will focus on geometric Proportional Analogies and their use in solving proportional problems as presented in Intelligence Tests. An exemplary task is presented in figure 1. In this paper, the analogical reasoning problem to be solved is defined as follows: Given as input A is related to B as C to some X, output that X (Kedar-Cabelli, 1988).

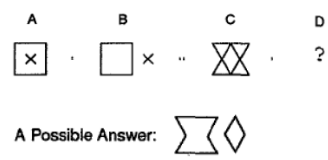

Fig. 1. Example for a proportional analogy task (O'Hara & Indurkhya, 1994).

2.2 Analogical Reasoning

Analogical Reasoning is a specific conclusion method. It is based on the connection between the representation of two situations. The well-known source domain's knowledge is transferred to the unfamiliar target domain to explain or predict unknown aspects in the target domain or to solve problems (Wenke, 2006).

The general operation principle is the following:

- Retrieval of known problems and their solutions
- Mapping of base and target domain

It can be described as back referencing to already known and solved problems and usage of experience in the base domain, thus, it is the transfer of something known to something unknown.

In general, Gentner proposed five sub processes of Analogical Reasoning in human thinking (Gentner, 1983, 1989):

1. *Representation:*
   Structural representation of the problem as graph, term, semantic net etc. This representation is essential for the mapping success. Syntactic mapping in bad representations can be impossible (e.g. What if one domain holds the relation A above B and the other domain holds the relation C under D? In this case it is impossible to map C to B and A to D, even if their relations mean the same).
2. *Retrieval:*
   Retrieval means to give access to already gained knowledge. Here, retrieval of an appropriate, already known problem is necessary. For successful retrieval it is relevant to get to know one problem in different domains to learn which features and relations are important and which are irrelevant.
3. *Mapping:*
   The Mapping of the known and the current problem is the core element of Analogy. It includes mapping of relations and functions of base and target domain.
4. *Inference/Transfer:*
   The generation of a solution is based on the mapping. It is a carry-over of information from base to target domain.
5. *Learning:*
   Generalization over problems leads to achievement of new knowledge through acquisition of more general schemes or rules by abstraction.

2.3 Case-Based Reasoning vs. Analogical Reasoning

On the one hand, studies in this field of research often tend to use case-based reasoning and analogical reasoning as synonyms. On the other hand, there are also differences between those two terms. Firstly, analogical-reasoning is used to characterize methods solving new problems based on past cases from a different domain, while case-based reasoning typically focusses on indexing and matching strategies from the same domain (single domain cases). Thereby, it is possible to consider analogical reasoning as a subfield of case-based reasoning concerned with mechanisms for identification and utilization of cross-domain transfers (Aamodt & Plaza, 1994).

Furthermore, case-based reasoning can be considered as a form of intra domain analogy, thus, staying in one domain within which a similar case is searched. Analogical reasoning in problem solving, on the other hand, is conducted between different domains (a similar case is mapped from the base to the target domain, as already stated above). Therefore, this distinction opposes the one made earlier, seeing analogical reasoning as the more general one (Aamodt & Plaza, 1994).

Another difference lies in the orientation of the two conclusion methods. While Analogical Reasoning focusses on abstract knowledge and structural similarity, Case-Based Reasoning emphasizes the relations between specific episodes and is more pragmatic oriented (Betsch, Funke, & Plessner, 2011). Case-Based

Reaonsing does not abstract to general rules, it mainly focusses on one concrete case.

Furthermore, case-based reasoning is feature based, whereas analogical reasoning is structure based and therefore more general and closer to the abstraction level (Gentner, 1983; Gentner & Markman, 1997).

Overall, this field's of research focus is the understanding and modelling of the reuse of past cases, called the mapping problem. The goal is to find a way to transfer or map the solution of an identified analogue source base to the present problem (target base).

All in all, there are different ways to differentiate between case-based and analogical reasoning, which are not uniformly used in literature.

2.4 Reasoning vs. Problem Solving

Briefly explained, reasoning is the transfer of one rule from one domain to another (e.g. the transfer of one relation in Proportional Analogies), whereas problem solving means that a sequence of rules is transferred to solve a problem (DeLoache et al., 1998).

This paper's focus lies on solving geometric proportional problems by applying analogical reasoning, thus, simple problems can be solved using reasoning.

2.5 The Importance of Analogical Reasoning in AI

As Analogical Reasoning is central to intelligence, and it is the aim of AI to create intelligent machines, the importance of Analogical Reasoning for AI should be clear. The inability to exploit past experience in solving problems in current expert systems is one bottleneck to truly built robust and intelligent systems (Kedar-Cabelli, 1988).

Additionally, Analogical Reasoning is necessary in AI systems to comprehend metaphors when understanding natural language, to learn new concepts by analogy similar to concepts in machine learning and to create computers with commonsense reasoning (Kedar-Cabelli, 1988).

As already mentioned, Analogical Reasoning in AI is one approach used in machine learning. In contrast to most of the other approaches in Analogical Reasoning not the entire problem space is searched, but the set of already known, similiar problems with known solutions is covered by the search. It is a search for the appropriate example in the example space to transfer the known solution to the current problem.

As in human analogical reasoning the computational analogical reasoning process has main components (Hall, 1989):

1. *Recognition:* A candidate for the source domain must be recognized, given the description of the target.
2. *Elaboration:* of the mapping between target and source domain.
3. *Evaluation:* of the mapping and the resulting inference in the use context, including justification, repair and extension of the mapping.

4. *Consolidation:* of the analogy's outcome so that its results can be sensibly reused in other contexts.

Important for every computational approach for producing analogical reasoning is the adequate representation of the basic materials of analogy: source domain, target domain, analogical inference and confirmatory support (Hall, 1989). Concretely said, a good representation of the source as well as of the target domain is needed to enable the recognition of analogical sources. Furthermore, evaluation of analogical inference and consolidation of confirmed inferences are essential. This leads to the necessity of processes for accessing, manipulating and creating these representations (Hall, 1989). For reasoning and learning these representations and processes are integrated into existing computational accounts (Hall, 1989).

3 A Basic Approach to Geometric Proportional Analogies

The following paragraph deals with the approach by Evans (1964). His Program ANALOGY was built to solve Geometric Analogical Reasoning Tasks as used in intelligence tests.

3.1 An Overview of Analogy

Evans (1964) was the first to try to solve geometric proportional analogies with a computer program. The resulting program called Analogy can solve geometric analogy tasks as used intelligence tests.

One Example for this kind of task is: Figure A is to figure B as figure C to which of the following five figures? (see figure 2).

The task is to find the rule which changes figure A into figure B, to apply this rule to figure C and to select the resulting figure from the possible answer figures 1 to 5. Evans (1964) wrote his program in LISP. It uses heuristic methods to calculate complex descriptions of figures from relatively primitive inputs and to use this descriptions in finding appropriate transformation rules as well as in applying these rules.

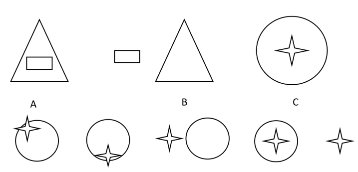

Fig. 2. Input Task for Analogy.

3.2 A Rough Sequence of Analogy

To constitute the exact program execution clear, the example above is used subsequently (see figure 2). Firstly, a problem, in this case one geometric analogy task, is selected. The input to Analogy are primitive descriptions in the form of dots and lines of in total eight figures. Analogy decomposes this descriptions into constituent objects. The result includes objects and binary relations (e.g. LEFT, ABOVE, INSIDE) over those objects within each figure (Evans, 1964; Hall, 1989). The figures labelled A, B and C are actual input figures, the five figures labelled from 1 to 5 are the different response options in the task. Now every figure is parsed in separate objects. The features of these objects as well as the relations among them are calculated using a heuristic approach, as already mentioned. Given this higher-level description Analogy now solves the problem in four steps:

- *Step 1* Generate a sequence of rules that transforms figure A into figure B.
- *Step 2* Generate a sequence of rules how figure C can be transformed into each if the given answer figures.
- *Step 3* Compare the rule sequence that leads from A to B with each sequence that leads to an answer figure.
- *Step 4* Select the rule sequence that transforms C into one answer figure and preserves the post information about the A-to-B rule.

Output is the number of the response option, that fits the generated solution. If no solution is found, the user is notified.

The rules are generated by a restricted mapping between two figures. This mapping is only possible, if both objects are of the same geometric type (meaning they must have the same shape) (Evans, 1964). To deal with differences in objects' size or orientation, Euclidian distance transformations (like scale change, rotation and reflection) are introduced. Deleting or adding an object in one figure that can be mapped is forbidden, but if one object fails to map with any object in another figure, it can be either deleted or added (Hall, 1989).

Therefore, a rule used in Analogy has three components (sets): objects that are deleted, objects that are added and objects that are mapped (Hall, 1989).

4 Fluid Geometrical Analogies: Reinterpretation and Rerepresentation

Subsequently, the approach of O'Hara and Indurkhya (1994) on integrating reinterpretation in analogical reasoning shall be explained and briefly discussed.

4.1 Reinterpretation

Generally, the term reinterpretation means that something is explained or defined in a new, different way. Within the context of Analogical Reasoning it means the ability to see one task in different ways. Particularly, in the context of proportional geometric analogies the capacity to see different ways to construct one figure is crucial. To illustrate this, the following example is given: The star of David is a quite popular figure, most of the people tend to claim that it is constructed of two triangles, as it is shown at the left side in figure 3. But, it is also possible to construct the star of David by combining three diamonds, as at the right side in figure 3. The outstanding capability of the human to see different construction options to result in one figure is a classic example for reinterpretation. To be able to solve more complex geometric analogies, it is important for AI systems to dispose this ability as well. For enabling this switch in point of view O'Hara (1992) introduced an algebraic formulation of the re-description process in the context of geometric proportional analogies. Therefore, reinterpretation in terms of AI systems means rerepresentation of the problem in a different way. This is used in the system PAN, which is described subsequently.

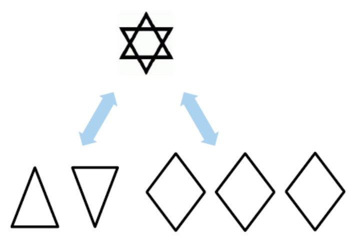

Fig. 3. Reinterpretation in case of the construction of the Star of David.

4.2 An Overview of PAN

The first approach to integrate reinterpretation in an analogical reasoning system was proposed by O'Hara and Indurkhya (1994). They managed to solve geometric proportional analogies with their program called Proportional ANalogy (PAN). The input for PAN are three figures A, B and C in form of straight line segments. The output is a figure D, such that the four figures A, B, C and D satisfy the proportional analogy relation: A is to B as C to D. PAN itself generates more complex descriptions on a higher conceptual level of the figures during the program execution (Hall, 1989). This involves rotations, translations, repetitions, symmetry etc. This is one of the key features of PAN as it is able to construct those figure descriptions "in context" (O'Hara & Indurkhya, 1994), which means the figures and corresponding descriptions function as context for the others during their construction.

4.3 The Architecture of PAN

PAN consists of different parts which shall be introduced in this paragraph. An overview about the whole architecture is given in figure 4. Starting from the bottom with the Raw Figure Data (a line segment set that represents the input figures A, B and C) PAN begins its performance. Firstly, the Preprocessor module is activated and converts the raw input data into a graph-like structure to guarantee an efficient processing of the data. For each figure, one graph is constructed and stored in the Description Space. The Description Space is the internal data base of PAN. Now the preprocessing step is completed and PAN starts a search process to build the already mentioned descriptions of the figures in the description space. Initially, all figures have no descriptions. Iteratively the descriptions of the figures in the description space are built through the following steps:

1. Investigation of the already existent descriptions in the description space.
2. Selection of one action to be taken in the description space (Proportional Analaogy or Perception module springs into action).
3. Applying the action in the description space.

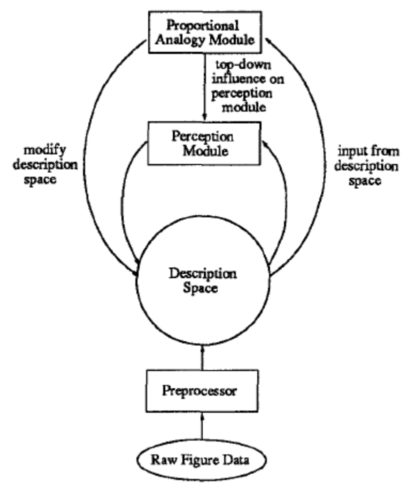

Fig. 4. The Architecture of PAN (oval shapes represent data structures, rectangular shapes represent processes)
(O'Hara & Indurkhya, 1994).

These steps are executed by the two blocks called Perception Module and Proportional Analogy (PA) Module. These two modules carry out modifications in the description space, represented in figure 4 by the arrows pointing towards the description space.

The general role of the Perception module is to find geometric concepts or subfigures within the input figures. This is done by various perception modules within the big perception module that detect different kinds of geometrical concepts (i.e. polygons, partial polygons, iterated or repeated figures and relations between the figures). As there might be a huge number of geometrical concepts within one figure, an order is imposed for detecting the concepts. In general, simple concepts and shapes are detected first, e.g. in the case of polygons first regular ones are detected followed by polygons with fewer regularities (O'Hara & Indurkhya, 1994).

The Proportional Analogy Module on the other hand, forms the actual proportional analogies. It is like a single perception module just for detecting analogies and consists of a set of condition-action rules for building analogies. The condition part of one rule is applied to the description space where it detects relationships between the partial descriptions of the figure that are relevant for creating a proportional analogy. The action part of the rule modifies the description space, if the condition of this rule is mapped to existing descriptions, to build proportional analogies. The actions the PA module can perform are diverse and include: activating the perception module to detect new geometric concepts in one or more of the figures, extending one or more partial descriptions of the figures A, B and C, creating mappings between partial descriptions of these figures, evaluating the quality of the proportional analogy that has been constructed so far and constructing figure D when the descriptions of A, B and C are complete.

4.4 A Sequence of PAN

PAN uses a greedy multiple hill-climbing search strategy, thus, the search focuses on building one single proportional analogy within the search tree at any time. The next step in creating this proportional analogy is chosen in a hill-climbing fashion which means the current best solution in the search tree is selected. But how can PAN compare different proportional analogies and select the best one? To make evaluation of analogies possible O'Hara and Indurkhya (1994) introduced the Proportional Analogy Quality (PAQ-) value. The PAQ-function is defined in the integer scale from zero to infinity where a value of zero stands for the best analogy. One analogy is supposed to be better than another one involving the same figures if its PAQ-value is smaller. Shortly said: The smaller the PAQ-value the better the analogy. During the search process one PAQ-value is assigned to each constructed analogy. How this value is formed concretely is described in the paper of O'Hara and Indurkhya (1994), and will be omitted in this paper for its limited content. In the beginning of the search, three fixed parameters are defined: The optimal quality threshold, the minimum quality threshold, and the search tree size. Those parameters function as termination criteria. The optimal quality threshold specifies a maximum PAQ-value below which the search may stop because an appropriate analogy was found. On the other hand, the minimum quality threshold defines the PAQ-value that an acceptable analogy can maximally have, above this value a solution is not even considered. If this value is exceeded the search is aborted and no appropriate solution is found. Lastly, a limit on the search tree size is defined limiting the search space as well.

With these parameters it is guaranteed that the progress along the current search path will stop if either a proportional analogy is found, no further progress can be made from this node within the allowed search tree size or the minimum quality threshold is exceeded. If the progress terminates in the current path, the search must be continued from one of the other nodes in the search tree, given the case no appropriate analogy was found. As the whole search tree consists of partially constructed proportional analogies each node holds one each. Criteria for choosing the next considered node are determined empirically, so that the new node holds the currently most developed proportional analogy with the lowest PAQ-value. If the entire search tree is searched without finding a proportional analogy that satisfies the optimal quality threshold, the analogy with the so far lowest PAQ-value is simply returned.

To show how the example of figure 1 can be solved using PAN, figure 5 shows one possible search tree, which shall be explained in the next paragraph.

The search starts at node 0 at the top. Here, the three input figures A, B, and C are presented. PAN starts to focus on one of the three input figures (in this case figure C). Now, the first step is to search for the primitive subfigures with the lowest PAQ-value that can be transferred into the input figure by applying some operations. In this case, the four small triangles might be found. Then there could be a preference of higher over lower figures, so that the two upper triangles are left. Finally, left hand figures could be preferred over right hand figures ending with the small triangle in the upper left corner as shown in figure 5, node 1 as primitive subfigure for constructing figure C. In node 2, figure 5 the process of applying various operations on this simple geometric figure starts. In the third node (figure 5), a similiar approach is used to form the final reflective iterative process for constructing figure C. This results in a full description of figure C. Now, the PA module starts with constructing the descriptions of the two other figures. Since figure C must be a result of adaptions applied to A's descriptions (this holds because of the underlying relation A is to B as C to X), the focus now shifts to building A's description. PAN now tries to build the description of A structurally similar to the description of B, which can be seen in node 3 (figure 5). If the rebuilding of A similar to C is successsful this will result in a lower overall PAQ-value of the constructed proportional analogy. After the third step in this example, the minimum quality threshold is most likely exceeded and the search is terminated. Thus, a new start point has to be found. So now, PAN starts again from node 0 (figure 5) with decomposing another figure (in this case: figure A) into subfigures. Here, PAN detects a square and a cross as primitive subfigures. These figures have no connections which makes the task easier. As A cannot be decomposed in more subfigures PAN decides to join the two subfigures with the INSIDE operator. After successfully constructing figure A (figure 5, node 5), PAN proceeds with the search for a way to construct figure B in node 6 (figure 5). The fact that A could be decomposed to a square and a cross in its description is directly used as starting point for the description of figure B which is decomposed in a square and a cross as well. The found description of B is pointed out as a square left of a cross. This adaption has a low PAQ-value as no subfigures are inserted or deleted. Also, the mapping costs from cross to cross, square to square and INSIDE to LEFT are low. In the next step, the found operations on how to build the figures A and B are applied to structure figure C. Here, the context of A's description is used to influence this process. The INSIDE operator in A's description causes the system to look for containing and contained subfigures. This results in choosing the outer concave polygon in figure C. Hence, in node 7 (figure 5), a full description of C is created. Now, the important reinterpretation and rerepresentation step is visible because this time figure C is constructed in a different way from different geometrical base figures. With the three complete descriptions of the input figures A, B and C and the adaptions that cause the transfer from figure A to figure B, finally, a solution figure D is constructed accordingly, which is given back as output.

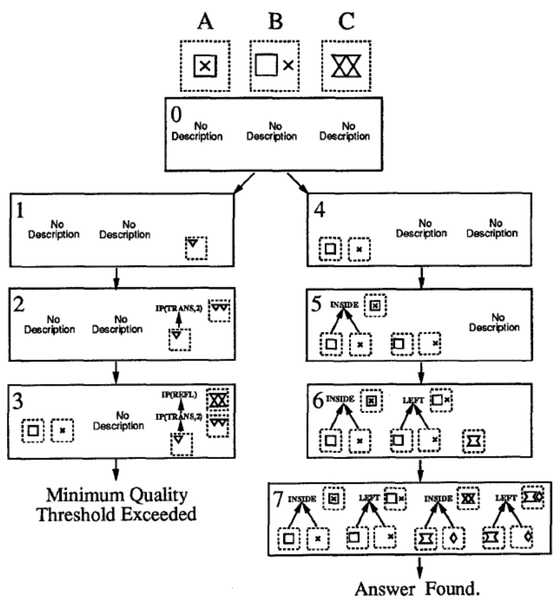

Fig. 5. Solving a geometric proportional analogy task with PAN (O'Hara & Indurkhya, 1994).

5 Critical Discussion

Finally, a discussion of strengths and limitations regarding Analogical Reasoning in AI research will be given. Firstly, the approach by Evans (1964) shall be briefly discussed in the following paragraph. All in all, Evans (1964) focuses on elaborating an analogy between two representations from the same problem domain. Therefore, elements are mapped at two levels: direct object mapping during rule generation and object role mapping during rule comparison (Hall, 1989). This results in constrains on both levels. On one hand, the mapped objects are required to be of the same type, on the other hand, the mapped objects have to play a similar role, i.e. deletion, addition or mapping (constrain of the second level) (Hall, 1989).

Moreover, in the case different rules are leading from A to B, but applying them on C would lead to different Ds, the analogy might be solved correctly, but the resulting figure might not be among the answer figures, which will lead to the response that no analogy was found, as well. Thus, Analogy is restricted by the given answer options. Reviewing the approach by Evans (1964), it needs to be said that according to the recognition of a candidate, the analogy is solved by giving the reasoner a source analogy, which is the most effective but least simply solution (Hall, 1989). Regarding the elaboration of the analogical mapping, especially the constraints on analogical mapping, Evans (1964) deals with a restriction of a one-to-one mapping of the rule components. With regard to the varieties of analogical inference it has to be said that Evans (1964) Analogy generates a set of generalized rule candidates, choosing the one that best preserves a one-to-one, type-consistent mapping between source figures (Hall, 1989). This causes that Analogy is not able to find solutions to problems that require transformations of relations as well as of objects (e.g. ABOVE relation in A:B should correspond to a LEFT relation in the preferred choice). The system would drop these relations during generalisation, so a correct discrimination of the ranked candidates is not possible (Hall, 1989).

The biggest point of criticism while discussing the approach of O'Hara and Indurkhya (1994) is that the architecture of their system PAN leads to great computational effort as it is a computationally expensive process. Regarding problems in practice there is not much to say as the paper states the status of PAN when it was not yet finished (O'Hara & Indurkhya, 1994). But still it is clearly pointed out that a lot more research is needed in the field of reinterpertation in analogical reasoning to make it useable in practice and to gain a domain-independent analogical reasoning module. For this purpose, the mechanisms of reinterpretation have to be studied in a number of different domains in which analogical reasoning system already exists, then reinterpretation modules have to be designed and implemented. Only after that it might be possible to develop a domain-independent analogical reasoning module (O'Hara & Indurkhya, 1994).

One important point of criticism is that both approaches to computational analogical reasoning do not provide constraints on which aspects of the source should be extended to the target (Kedar-Cabelli, 1988). In most cases, contextually-relevant analogical inferences are just a small subset of all possible inferences (Greiner, 1988). In order to counter this, Kedar-Cabelli (1986) presented a model that uses knowledge of the purpose for which an analogy is being constructed to constrain the process of elaboration. This makes the process more sufficient and realistic. To put it shortly: the here described problem depends on the knowledge representation within one computational approach of solving analogies. According to the used representation of base and target domain, some kinds of analogies might be better or faster solved than others, depending on which features are encoded and if these are useful for building the proportional analogy.

Lastly, the relation of analogical reasoning in geometrical context and spatial thinking shall be mentioned. One point of criticism by reviewing the approaches on building geometric proportional analogies might be the question, if not the ability to solve analogical reasoning problems but the ability of spatial thinking is tested and needed. This question seems legitimate as the ability to encode and transfer geometric objects and to determine subfigures within the given figures is crucial to solve those problems. But still, this ability is just the precondition to solve proportional analogies, not the ability itself, as there are more processes needed to solve those, as presented in the chapters before.

6 Conclusion

As this paper tried to show, there are some approaches dealing with the implementation of systems that are able to solve geometric proportional analogies. All of them have different advantages and disadvantages.

The system Analogy by Evans (1964) was presented and explained. It is important to note that it is the first approach dealing with geometric proportional analogies in a computational way.

After this, the attempt to incorporate reinterpretation in analogical reasoning by O'Hara and Indurkhya (1994) was introduced. This approach called PAN is dealing with geometric proportional analogies, as well. But now, one figure can be represented in different ways, i.e. a composition in different subfigures is possible, leading to different possible proportional analogies.

In a nutshell, the topic analogical reasoning is still very important in our society and it is therefore critical to conduct more research in this area to fully understand the underlying mechanisms and to make them usable in the field of AI. In the future, it is of great importance to support further research in this field.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*(1), 39–59.

Betsch, T., Funke, J., & Plessner, H. (2011). *Allgemeine Psychologie für Bachelor: Denken-Urteilen, Entscheiden, Problemlösen.* Springer-Verlag.

DeLoache, J. S., Miller, K. F., & Pierroutsakos, S. L. (1998). Reasoning and problem solving.

Evans, T. G. (1964). A heuristic program to solve geometric-analogy problems. In *Proceedings of the April 21-23, 1964, Spring Joint Computer Conference* (pp. 327–338).

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155–170.

Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 199–241). Cambridge University Press.

Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *American Psychologist*, *52*(1), 45–56.

Greiner, R. (1988). Learning by understanding analogies. *Artificial Intelligence*, *35*(1), 81–125.

Hall, R. P. (1989). Compuational approaches to analogical reasoning: A comparative analysis. *Artificial Intelligence*, *39*(1), 39–120.

Kedar-Cabelli, S. (1986). Purpose-directed analogy: A summary of current research. In *Machine Learning* (pp. 123–126). Springer.

Kedar-Cabelli, S. (1988). Analogy—from a unified perspective. In *Analogical Reasoning* (pp. 65–103). Springer.

O'Hara, S. (1992). A model of the 'redescription' process in the context of geometric proportional analogy problems. In *International Workshop on Analogical and Inductive Inference* (pp. 268–293).

O'Hara, S., & Indurkhya, B. (1994). Incorporating (re)-interpretation in case-based reasoning. In J. G. Carbonell et al. (Eds.), *Topics in case-based reasoning* (Vol. 837, pp. 246–260). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-58330-091

Schmid, U., Gust, H., Kühnberger, K.-U., & Burghardt, J. (2003). An algebraic framework for solving proportional and predictive analogies. In *Proceedings of the European Conference on Cognitive Science (EuroCogSci 2003)* (pp. 295–300).

Wenke, D. (2006). Analoges Schließen. In J. Funke & P. A. Frensch (Eds.), *Handbuch der Allgemeinen Psychologie-Kognition* (pp. 406–411). Hogrefe Verlag.