

Transformational and derivational strategies in analogical problem solving

Sven-Eric Schelhorn · Jacqueline Griego ·
Ute Schmid

Received: 24 July 2006 / Revised: 30 November 2006 / Accepted: 4 December 2006 / Published online: 19 January 2007
© Marta Olivetti Belardinelli and Springer-Verlag 2007

Abstract Analogical problem solving is mostly described as transfer of a source solution to a target problem based on the structural correspondences (mapping) between source and target. Derivational analogy (Carbonell, Machine learning: an artificial intelligence approach Los Altos. Morgan Kaufmann, 1986) proposes an alternative view: a target problem is solved by replaying a remembered problem-solving episode. Thus, the experience with the source problem is used to guide the search for the target solution by applying the same solution technique rather than by transferring the complete solution. We report an empirical study using the path finding problems presented in Novick and Hmelo (*J Exp Psychol Learn Mem Cogn* 20:1296–1321, 1994) as material. We show that both transformational and derivational analogy are problem-solving strategies realized by human problem solvers. Which strategy is evoked in a given problem-solving context depends on the constraints guiding object-to-object mapping between source and target problem. Specifically, if constraints facilitating

mapping are available, subjects are more likely to employ a transformational strategy, otherwise they are more likely to use a derivational strategy.

Introduction

Analogical reasoning and knowledge transfer are long recognized as core mechanisms in education, psychology, cognitive science, and machine learning. For example, students typically learn math concepts by example and are tested on new problems using the same concept. Medical students use an example heart disease case to diagnose new patients. In everyday life a person might use their knowledge of train travel when planning a trip by air.

In cognitive theories analogical transfer of knowledge is typically modeled as transformation of the source solution based on the mapping of source and target structure (e.g., SME, Falkenhainer et al. 1989; LISA, Hummel and Holyoak 1997). Carbonell (1986) characterizes the problem solving strategy described by these structure mapping based models as transformational analogy and contrasts it with derivational analogy, which he proposed as a computational approach to analogy in the context of artificial intelligence planning systems. Using a derivational strategy, a problem-solving episode consists of the reasoning traces (derivations) of past solution processes, including the explored sub-goal structure and methods used.

Carbonell argues that, from a computational point of view, transformational analogy is often inefficient and can result in suboptimal solutions. Instead of

S.-E. Schelhorn
Institute of Cognitive Science, University of Osnabrueck,
Osnabrueck, Germany
e-mail: sven-eric.schelhorn@uos.de

J. Griego
School of Medicine, Department of Psychiatry,
University of Maryland, Baltimore, USA
e-mail: jgriego@mprc.umaryland.edu

U. Schmid (✉)
Information Systems and Applied Computer Science,
Otto-Friedrich University Bamberg,
Feldkirchenstrasse 21, 96045 Bamberg, Germany
e-mail: ute.schmid@wiai.uni-bamberg.de

calculating a source/target mapping and solving the target by re-instantiation of the source solution, it might be more efficient to reconstruct the solution process; that is, use a remembered problem solving episode as guideline for solving the new problem.

Derivational analogy can be characterized by the following sub-processes: (a) retrieving the problem solving episode which shares the most top-level goals with the new problem, (b) applying the retrieved derivation to the current situation by ‘replaying’ the problem solving episode, and (c) checking at each step whether the derivation is still applicable in the new problem solving context (Carbonell 1986; Veloso and Carbonell 1993). This is in contrast to the transformational sub-processes: representing the target problem, retrieval of a previously solved source problem, mapping of the structures of source and target, transfer of the source solution to the target problem, and generalizing over the common structure of source and target (e.g., Keene et al. 1994).

We see one superficial and one crucial difference between the transformational and the derivational approach: the first and crucial difference is the different assumptions concerning the process underlying analogical transfer made by the approaches. Models of transformational analogy postulate that the internal structure of relations of a source problem is mapped to the target and the relevant substructure is transferred. Derivational analogy assumes that a problem solving technique that was successful for a source problem is applied in the new context and replayed step-by-step. In either case knowledge from the source problem must be mapped to the target problem. However, in transformational analogy inferences from the source relational structure are then “projected” onto the target (Gentner 1983), while in derivational analogy inferences learned from the source solution are unnecessary. This makes derivational analogy especially useful for problems where a relational structure mapping is problematic. A computational example is translating Pascal code into Lisp where a line-by-line translation is problematic because the languages’ relational structures are different (Carbonell 1986). Overall, neither strategy dominates with respect to information processing efficiency.

Previous results (e.g., Holyoak and Thagard 1989) highlight how pragmatic constraints, such as goals, produce pressures towards a type of solution strategy. For example, an algebra word problem might be solved faster by mapping and transfer than by derivation, with the danger that the problem is correctly solved but not understood (Novick and Hmelo 1994). Typically, in algebra word problems, numerical amounts are

strongly associated with objects playing specific roles in the context of a problem. If there are corresponding roles in base and target problems, mapping is facilitated by the semantic constraints introduced via the roles. On the other hand, calculating a shortest route in a maze (Novick and Hmelo 1994) can be solved faster by just applying a breadth-first search strategy learned when solving the problem for a different maze than by trying to map the mazes because there are no clear constraints to indicate which intersection in a base maze should be mapped to which intersection in the target.

The second and only superficial difference between transformational and derivational analogy is the elaborateness of the representations. For transformational analogy, it is usually assumed that only the problem definition and its solution are represented; in derivational analogy, complete problem solving episodes, regarding each problem solving step, are stored in and retrieved from memory. It is an open question what kind of information is retained by humans. It may be more than only the solution, but less than every solution step together with its justifications. The important point is that memory of a successful solution procedure is a feature of derivational analogy, while explicit aspects of the solution procedure are not implemented in transformational analogy. Full derivational storage is a reasonable approach for many artificial intelligence (AI) applications for which the derivational approach was developed. However, we are not proposing the “store everything” extreme as a cognitive model.

Although derivational analogy was first introduced as an AI model, it seems to be a plausible model for describing human problem solving. For example, Catrambone and Holyoak (1989) demonstrate that individuals transfer sub-goals and methods to new problems rather than mapping the complete solution. Ormerod et al. (1990) showed spontaneous strategic transfer, as opposed to mapping-based transfer, in the domain of Prolog programming. Schmid and Carbonell (1999) show that in graph-search problems individuals exhibit a derivational rather than a transformational strategy.

In previous work (Schmid and Carbonell 1999) we demonstrated, in contrast to standard findings where potential problem solutions are selected on the basis of surface features such as object similarity (cf. Novick 1988), if subjects first try to solve a target problem, then they will select a second “helpful” problem based on the best match out of the potential source solution procedures. That is, they had a concept of the solution procedure and could recognize it as important for another problem solution (cf. Reed et al. 1990).

Additionally, we found that derivational analogy can support understanding a new domain and learning general solution procedures via the modular components (i.e., sequence subcomponents) of the derivational process (Gerjets et al. 2002).

Considering these findings, we propose that both transformational and derivational analogy were problem-solving strategies realized by human problem solvers. Which strategy is evoked in a given problem solving context depends on (a) the particular problem (low numbers of objects in the problem or high semantic similarity between objects of old (source) and new (target) problems give advantage to TA), (b) the degree of surface similarity between old problems retrieved from memory and new problems (high surface similarity evokes TA), and (c) the problem solving goal (e.g., fast solution vs. understanding and learning).

As a first attempt to identify analogical problem characteristics that bias evoking either transformational or derivational analogy, we focus on the ease of object-to-object mapping, more concretely, on the number of constraints available to establish correspondences between objects from the source and the target domain. In the following, we present the domain of path-finding problems where it is possible to vary the similarity between old and new problems in a systematic way. Afterwards, we present an experiment, extending the study of Schmid and Carbonell (1999), investigating whether perceived object-to-object correspondences between source and target trigger a transformational strategy, while in absence of correspondence cues, derivational analogy is evoked.

Path-finding problems

To empirically investigate strategy use depending on mapping constraints we employed variants of the path-finding problems presented in Novick and Hmelo (1994), namely Eulerian trails. These problems can be solved by finding a path through a graph beginning at a given node and using each arc exactly once. An example problem is given in Fig. 1.

To find an order in which people pass messages between each other, one can first draw a graph where each person introduced in the text is represented by a node and each (symmetrical) acquaintance relation is represented by an arc (see Fig. 2). Given the graph, a correct sequence can be found by starting with the node representing the starting person indicated in the problem description (here “S”) and then trying to find a legal sequence of arcs. For the graph depicted in Fig. 2, which has the well-known “envelope” structure,

Birthday. Five people attended a birthday party: Richard, Eric, Mary, Susan, and Bill. During the course of the evening they played different games. One game they played was a “messenger game” where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Susan, Eric and Bill all know each other; Richard and Susan know each other; Bill, Mary, and Richard all know each other; Eric and Richard know each other. Susan was the person writing the first word. Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as “S” for “Susan” and so forth.

Fig. 1 Birthday problem (low guidance of mapping version)

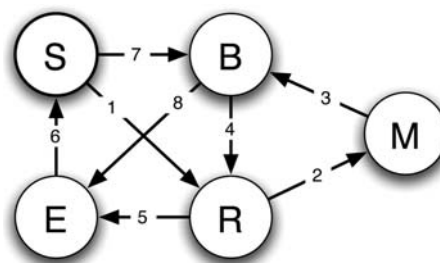


Fig. 2 Example Eulerian trail

there exist 44 legal solutions. From an algorithmic point of view, there exist efficient procedures finding a Eulerian trail (Christofides 1975). One legal solution is given in Fig. 2.

Graph problems of this kind are especially suited to test the proposition that ease of mapping determines whether transformational or derivational analogy are adopted as problem solving strategy: for two problems whose underlying structure can be represented both by the same envelope graph (as the one given in Fig. 2) where in one problem five persons are represented as

nodes and in another five cities, there are next to no mapping constraints to map the graph structure from the first problem to the one of the second. The only constraint is the degree of a node (the number of arcs associated with it), which might be hard to detect for a person not experienced in graph theory. That is, if an analogical strategy is adopted, knowledge must be transferred without structural constraint mapping (specific for transformational analogy) through procedure re-instantiation alone (that is by derivation).

In contrast to such a low guidance of mapping relation between a source and a target graph problem, a high guidance of mapping relation can be realized by introducing additional mapping constraints, facilitating one-to-one mapping and thereby trigger the use of transformational analogy. The different objects represented as nodes could be given distinct roles (say father, mother, uncle, aunt, child) which occur in both base and target problem description. In our experiment, we did not introduce such semantic constraints, instead we used two “syntactic” constraints: (a) a surface similarity constraint where corresponding objects in source and target are named with labels starting with the same letter (e.g., person “Babs” and city “Behringen”) and (b) a structural constraint where the number of arcs associated with a node are explicitly stated in the same sequence in base and target problem. For example, the relation “node ‘a’ is only connected with ‘b’ and ‘c’” could be presented as “Babs knows two people—Ned and Mike”.

Pilot study

The two mapping constraints, which influence the guidance of mapping, were evaluated for saliency in a pilot using the methodology of Reed et al. (1990), in which they examined the effects of similarity and inclusiveness on mapping.

Individuals were given a version of a graph problem (see boat problems in Appendix 1) and instructed that they should assume they could not solve it. These were the sources for the analogy. They were then given two versions of another graph problem (see birthday problems in Appendix 1) that varied according to the availability of the surface similarity constraint (initial letter in object name) and the structural constraint (arc relations). Then subjects were asked which solution from the set of birthday problems they found better suited to use as an explanation for students in solving the boat problem. They were also asked which birthday problem version they found more similar and which they found more useful.

Subjects regularly rated the version of the second problem lower in usefulness for explanation and similarity that used none of the two mapping constraints introduced above. The version that made use of both constraints, on the other hand, was rated significantly higher. Therefore results of this pilot indicated that the surface similarity constraint and the structural constraint were indeed salient. These two mapping constraints were used for the high guidance of mapping materials in the main experiment.

Experiment

We conducted an experiment to test the assumptions that (a) in the absence of surface and structural constraints knowledge can still be transferred though procedure re-instantiation (using derivational analogy), and (b) that the derivation procedure in derivational analogy takes longer than mapping and transfer using transformational analogy, because the procedure evaluation and goal and sub-goal exploration are more time consuming than transferring the structure of the solution.

Method

Subjects

Subjects were recruited by an E-mail call sent to lists of undergraduate cognitive science students at the University of Osnabrueck, Germany, graduate psychology students at the University of New Mexico, and computer science students and graduates with various university affiliations (the mean age was 25.5 years).

A total of 51 subjects were randomly assigned to be in the high guidance of mapping condition or the low guidance of mapping conditions. Forty-two subjects had a correct mapping solution (see below), and all analyses were done with these data (30 males, 12 females). There was no significant difference in mean self-rated expertise (scale = 0–5) regarding graph problems in the high guidance of mapping group ($M = 2.95$; $SD = 1.29$) and the low guidance of mapping group ($M = 2.35$; $SD = 1.57$), $t(40) = 1.37$, $P = 0.18$.

Materials and apparatus

The materials were graph problems using Eulerian trails, as in Novick and Hmelo (1994). There were two problem types: “boat” and “birthday” (see Appendix 1).

Both can be represented as graph problems. The problem representation graphs are isomorphic, i.e., inherently identical. The only differences between the two stories are the setting. While “boat” deals with the problem of traversing all channels that connect five cities exactly one time, “birthday” deals with a message that should be passed between all friends exactly one time. This class of problems provides structurally similar problems with transferable solution procedures.

Between the two conditions, the “boat” problem and the “birthday” problem are varied in the degree of guidance of mapping they provide: A source/target pair with high guidance of mapping contains mapping constraints that supposedly simplify mappings between the problems and thus should trigger a transformational strategy. A source/target pair with low guidance of mapping, on the other hand, does not contain these constraints, and therefore supposedly is more difficult to map which should suppress transformational strategy and trigger the use of derivational processes.

Further, we developed a strategy assessment questionnaire (SAQ) for obtaining post hoc self-reports. The questionnaire has five forced-choice “yes” or “no” questions assessing transformational strategies, five questions assessing derivational strategies, and five filler questions for checking response reliability. Transformational questions expecting a “yes” response are focused at the concept of re-instantiation of a source’s relational structure; for example, “It was simple to use the boat solution to solve the birthday problem by replacing the names of the towns with the names of the people”. Derivational questions expecting a “yes” response are focused on the concept of re-instantiation of a solution procedure and the inability to directly transfer the base problem solution; for example, “I broke the birthday problem into smaller pieces and used the same strategies that were used in the boat problem solutions”. The full questionnaire is attached in [Appendix 2](#).

The experiment was conducted as a web-based implementation. Data collection was accomplished using a server running a guided set of PHP-webpages and a MySQL-database. Time measures were handled in the subject’s web-browser by a JavaScript-Function. Multiple submissions from the same participant were controlled.

Procedure

After a written procedural introduction, subjects read two example problems (The Fortress, Duncker 1945; Tumor, Gick and Holyoak 1983) and were given the “solution” in the form of an explanation of how the

problems are similar. This initial problem, together with verbal instructions, was introduced to induce subjects to employ analogical problem solving rather than problem solving from first principles. After indicating that they understood the solution, subjects then read the first experimental problem, “boat” (see [Appendix 1](#)). They were informed that a paper and pen is helpful for solving the problem, and were then taken through a written and visualized solution of the problem. Subjects were then instructed that they will see the second problem, “birthday”, and that they do not to have to memorize the first problem, as it will be available when needed. After reading the birthday problem they were instructed to solve it.

Subjects submitted their solutions to the birthday problem by entering the first letters of the nodes that were traversed in the graph, as it was introduced in the solution explanation for the first problem. They received no feedback about the solution. Subjects who did not solve this problem correctly were allowed to continue but were not included in the analysis of the study.

After that, the subjects were asked to indicate which city of the first (“boat”) problem corresponds to which person of the second (“birthday”) problem. Time was measured for this mapping. Again, they received no feedback about the solution.

Subjects then completed the SAQ. At the end of the experiment, gave demographic information and were thanked for their participation and given additional information regarding how to be informed about study results.

Results

The low guidance of mapping problems were not harder to solve than high guidance of mapping problems. The number of correct solutions between conditions was not significantly different, $\chi^2(1) = 0.19$. Thus, the lack of pragmatic (arcs) and syntactic similarity constraints (names) did not influence the correctness of the solutions.

The high guidance of mapping group mean solution time ($n = 22$, $M = 731$ s, $SD = 100$ s) was significantly faster than that of the low guidance of mapping group ($n = 20$, $M = 1,710$ s, $SD = 218$ s), $t(26.80) = 4.08$, $P < 0.001$ (see [Fig. 3](#)).

The SAQ results were analyzed using Mann–Whitney U -tests to evaluate whether both the mean ranked “yes” responses for the transformational scale, and the mean ranked “yes” responses for the derivational scale are dependent on the level of the agreement condition.

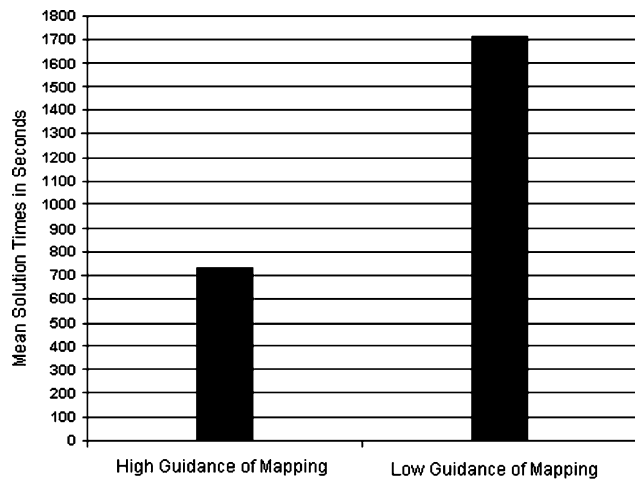


Fig. 3 Mean solution time

For the transformational scale the high guidance of mapping group ($n = 22$) had a significantly higher mean rank of 24.95 than the low guidance of mapping group ($n = 20$), which had a mean rank of 17.70, $z = 1.96$, $P = 0.05$. For the derivational scale the low guidance of mapping group had a significantly higher mean rank of 26.85 than the high guidance of mapping group (17.70), $z = 2.78$, $P = 0.005$ (see Fig. 4).

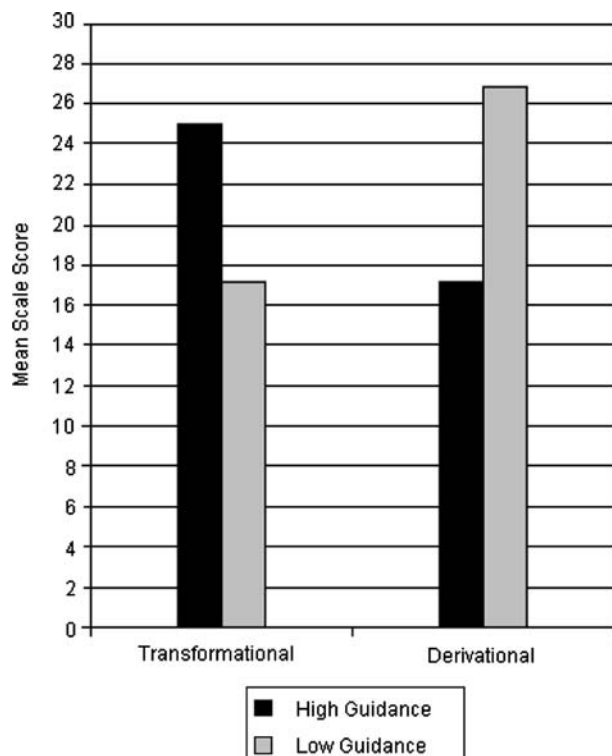


Fig. 4 Mean scale score

Thus, subjects from the high guidance of mapping group indicated significantly more use of transformational strategies than derivational strategies, and the low guidance of mapping group indicated significantly more use of derivational strategies than transformational strategies.

Additional scale properties were evaluated. Subject's scores on derivational items were negatively correlated with their scores on transformational scales $r(42) = -0.46$, $P = 0.002$. Therefore the SAQ exhibits a degree of dissociation in what is being measured. Furthermore, the correlation between mean solution time and mean derivational items score was positive, $r(42) = 0.37$, $P = 0.011$, which is expected under the assumption that derivational strategies involve more solution time. The relationship between mean solution time and the mean transformation items score was marginally significant in the negative direction, $r(42) = -0.27$, $P = 0.070$. Thus, higher transformational scores tended to be associated with faster solution times.

Self-reports of strategy are limited by the solvers' interpretation of the question and their degree of insight into what they did. Our tactic was to minimize interpretation effects with a forced choice of "yes" or "no" and to clearly define the concepts being measured. The questions were focused on properties we consider unique between transformational and derivational analogy: re-instantiation of a source's relational structure for transformational and re-instantiation of a solution procedure and the inability to directly transfer the base problem solution for derivational analogy. Additionally, these characteristics were assessed with five questions each, giving multiple measures of the same concept. This method is comparable to those assessing personality, intelligence, and mood.

Conclusion

Based on a study of Schmid and Carbonell (1999) where it was demonstrated that derivational analogy is not only a useful strategy for AI systems but is also applied by human problem solvers, we conducted an experiment to obtain more systematic information about the problem settings which trigger derivational in contrast to transformational analogy. We could show that mapping constraints play an important role in what kind of analogical strategy is adopted. Although subjects performed actions that were specifically related to a focus on a solution procedure, it was clearly demonstrated, in accordance with our theoretical

claim, that they used the transformational strategy when the mapping constraints were available, and the derivational strategy otherwise. That is, the path-finding problems did not unduly bias subjects towards a derivational strategy.

In follow-up experiments we plan to investigate further problem settings, which can influence the choice of analogical strategy. The most important aspect, in our opinion are differences in problem solving goals. We assume that—if mapping can be performed easily—transformational analogy can easily applied as a “lazy man’s strategy”, that is, a problem can be solved fast but without obtaining a real understanding of the domain. Consequently, an interesting experiment would be to induce “fast solution” versus “deep understanding” as problem solving goals and again assess whether this variation triggers transformational versus derivational strategy. Generalization learning should be assessed as additional dependent variable.

Thus far structure mapping has been the dominant mechanism in the exploration of analogical transfer, so solution procedures themselves are largely uninvestigated. We strongly believe that the AI approach of derivational analogy gives a fruitful theoretical perspective to work in psychology dealing with the distinct role of procedural knowledge transfer in contrast to structural transfer (e.g., Chen 2002; Novick 1988, 1990; Novick and Hmelo 1994; Sander and Richard 1997) which could also be exploited in the domain of instruction design as was, for example, shown in comparing transformational versus derivational worked-out examples with respect to transfer success in the domain of probability theory (Gerjets et al. 2002).

Acknowledgments We want to thank the two reviewers for helpful comments on an earlier version of this paper.

Appendix 1

Problem: low guidance of mapping

Source: boat

The Johnsons are planning a riverboat tour for their summer holiday visiting five cities: Schwetzingen, Marbach, Blaubeuren, Ludwigsburg and Ulm. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel

through any lock more than once. The eight locks are located between the following pairs of cities: Schwetzingen and Blaubeuren, Schwetzingen and Ludwigsburg, Schwetzingen and Marbach, Marbach and Ludwigsburg, Marbach and Blaubeuren, Blaubeuren and Ludwigsburg, Blaubeuren and Ulm, Ulm and Ludwigsburg. The Johnsons plan to start their trip in Schwetzingen. From Schwetzingen, they wish to travel along a route that will enable them to go through each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the cities, as “B” for “Blaubeuren” and so on.

Solution to boat

For solving this problem, it is very helpful to visualize it with the help of a pen and a sheet of paper. First note each city mentioned in the problem by its first letter. For example, to represent the city “Schwetzingen” you draw a capital “S”, for the city “Blaubeuren” a capital “B” and so forth. When you did this with all of the cities mentioned in the problem, your sheet will look similar to the one displayed below. Do not worry if it does not exactly like yours, because there are several correct possibilities.

Please refer to Fig. 5 for a representation of the solution step.

After you have represented the cities on your sheet of paper, let us proceed with the river locks between the cities. Each river lock lies on a channel that connects two cities. To represent this channel, and with it also the lock, draw a line between two cities on your sheet of paper. For example, if the text says that a lock is located between the cities “Schwetzingen” and “Blaubeuren”, draw a line between the capital letters “S” and “B” on your sheet of paper. After you have done this with all the connections mentioned in the problem, your sheet will look similar to this one.

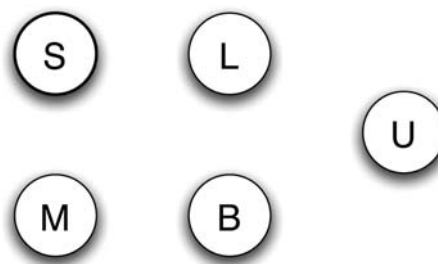


Fig. 5 Low guidance of mapping solution procedure step 1

Please refer to Fig. 6 for a representation of the solution step.

Now we have to plan a trip for the Johnsons, as it is mentioned in the problem. The Johnsons want to travel through every lock exactly one time. If we have our representation on the sheet of paper in mind, this means that we have to find a route, where every line is used exactly one time, starting with the city Schwetzingen. Each time we cross a city, we note its capital letter somewhere to keep track of our position. Of course we also have to keep in mind, which lines we already used, because we are not allowed to use them twice. At some place it might be, that our partial solution cannot be completed, because we made an error and end in a city without unused lines attached one to it. Then we have to go back and make another decision at an earlier point of the solving process. Let's have a closer look at one of the possible correct solutions. We start in Schwetzingen, as mentioned in the problem. From there, we travel over the line to Blaubeuren, further to Ulm, Ludwigsburg, again Blaubeuren, then Marbach, again Schwetzingen, Ludwigsburg and finally again Marbach. If the Johnsons follow this way, they will see each lock (that is, use each line) but no lock twice. Of course they travel through some cities twice, but that is not forbidden by the problem. So one possible solution to the problem would be "SBULBMSLM"—the order of visited cities the Johnson's could travel, represented by the first capital letters of the cities attached one to an another.

Please refer to Fig. 7 for a representation of the solution step.

Target: birthday

Five people attended a birthday party: Richard, Eric, Mary, Susan, and Bill. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on.

To make things not too simple, the message passing followed a complicated protocol: the message had to be passed between all people knowing each other, but was only allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Susan, Eric and Bill all know each other; Richard and Susan know each other; Bill, Mary, and Richard all know each other; Eric and Richard know each other. Susan was the person writing the first word. Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so forth.

Problem: high guidance of mapping

Source: boat

The Johnsons are planning a riverboat tour for their summer holiday visiting five cities: Cannenbach, Frankheim, Neustadt, Markburg and Behringen. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Behringen is connected to Neustadt and to Markburg via locks. These two cities are connected with all other cities via locks—that is, Neustadt is connected to Behringen, Markburg, Frankheim and Cannenbach; Markburg is connected with Behringen, Neustadt, Frankheim and Cannenbach. Frankheim and Cannenbach are connected with all cities except Behringen—that is, Frankheim is connected with Neustadt, Markburg and Cannenbach; Cannenbach is connected with Neustadt, Markburg and Frankheim. From Cannenbach, they wish to travel along a route that will

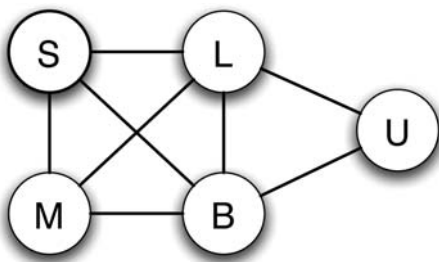


Fig. 6 Low guidance of mapping solution procedure step 2

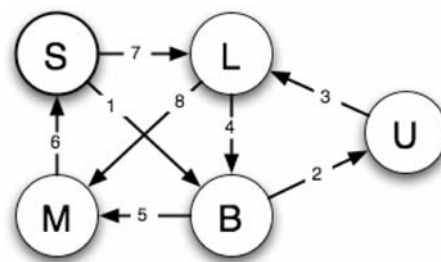


Fig. 7 Low guidance of mapping solution procedure step 3

enable them to go through each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the cities, as “N” for “Neustadt” and so on.

Solution to boat

For solving this problem, it is very helpful to visualize it with the help of a pen and a sheet of paper. First note each city mentioned in the problem by its first letter. For example, to represent the city “Cannenbach” you draw a capital “C”, for the city “Frankheim” a capital “F” and so forth. When you did this with all of the cities mentioned in the problem, your sheet will look similar to the one displayed below. Do not worry if it does not exactly like yours, because there are several correct possibilities.

Please refer to Fig. 8 for a representation of the solution step.

After you have represented the cities on your sheet of paper, let us proceed with the river locks between the cities. Each river lock lies on a channel that connects two cities. To represent this channel, and with it also the lock, draw a line between two cities on your sheet of paper. For example, if the text says that a lock is located between the cities “Behringen” and “Neustadt”, draw a line between the capital letters “B” and “N” on your sheet of paper. After you have done this with all the connections mentioned in the problem, your sheet will look similar to this one.

Please refer to Fig. 9 for a representation of the solution step.

Now we have to plan a trip for the Johnsons, as it is mentioned in the problem. The Johnsons want to travel through every lock exactly one time. If we have our representation on the sheet of paper in mind, this

means that we have to find a route, where every line is used exactly one time, starting with the city Cannenbach. Each time we cross a city, we note its capital letter somewhere to keep track of our position. Of course we also have to keep in mind, which lines we already used, because we are not allowed to use them twice. At some place it might be, that our partial solution cannot be completed, because we made an error and end in a city without unused lines attached one to it. Then we have to go back and make another decision at an earlier point of the solving process. Let’s have a closer look at one of the possible correct solutions. We start in Cannenbach, as mentioned in the problem. From there, we travel over the line to Neustadt, further to Behringen, Markburg, again Neustadt, then Frankheim, again Cannenbach, Markburg and finally again Frankheim. If the Johnsons follow this way, they will see each lock (that is, use each line) but no lock twice. Of course they travel through some cities twice, but that is not forbidden by the problem. So one possible solution to the problem would be “CNBMNFCMF”—the order of visited cities the Johnson’s could travel, represented by the first capital letters of the cities attached one to another.

Please refer to Fig. 10 for a representation of the solution step.

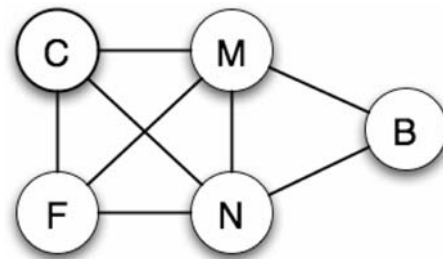


Fig. 9 High guidance of mapping solution procedure step 2

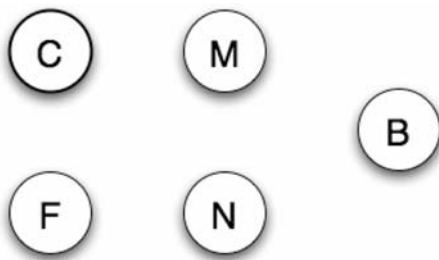


Fig. 8 High guidance of mapping solution procedure step 1

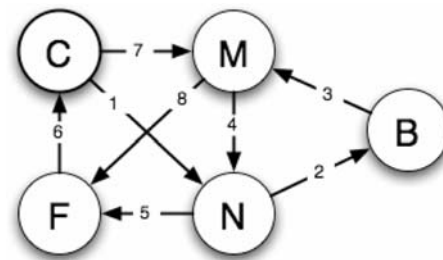


Fig. 10 High guidance of mapping solution procedure step 3

Target: birthday

Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a “messenger game” where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so forth. To make things not too simple, the message passing followed a complicated protocol: the message had to be passed between all people knowing each other, but was only allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Babs knows two people—Ned and Mike. Ned and Mike both know all other people—that is, Ned knows Babs, Mike, Fred, and Carry; Mike knows Babs, Ned, Fred, and Carry. Carry and Fred know all people except Babs—that is, Fred knows Ned, Mike, and Carry; Carry knows Ned, Mike and Fred. Carry was the person writing the first word. Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as “C” for “Carry” and so forth.

Appendix 2

Strategy assessment questionnaire

Remark

Questions designed to assess transformational strategies: 01, 04, 08, 10, 11, questions designed to assess derivational strategies: 02, 03, 06, 12, 13, all other questions were fillers for checking response reliability.

Instructions

Now you are nearly done. Please answer some final questions for us. After you answered all the questions, please click on “Submit Answers” on the bottom of this page. Describe how you solved the “birthday” problem. Check “yes” if a sentence approximately fits your strategy and “no” if it does not.

| | | | | | | |
|----|-----|-----------------------|----|-----------------------|---|---|
| 01 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | It was simple to use the “boat” solution to solve the “birthday” problem by replacing the names of the towns with the names of the people |
| 02 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | The “boat” problem and the “birthday” problem seemed similar but I could not figure out how the solutions were related |
| 03 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I remembered how I drew the graph with help of the solution of the “boat” problem. Through this I found the travel relations between the towns and used the same procedure to solve the “birthday” problem |
| 04 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I did not go through the steps I used in the “boat” problem to solve the “birthday” problem because I just replaced the names in the boat problem with the names in the birthday problem |
| 05 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I could not use the boat problem solution to solve the “birthday” problem, but when I finished with the “birthday” solution I realized how they are the same |
| 06 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I used the travel strategy I remembered from the boat problem to solve the “birthday” problem, but I made a few (or one) mistakes as I went along and had to do some of the message route over again |
| 07 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | The “boat” problem and the “birthday” problem seemed too dissimilar for me to use the “boat” solution to solve the birthday problem |
| 08 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I did not have to try to remember any of the steps from the “boat” problem solution |
| 09 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I could not solve the “birthday” problem |
| 10 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | The solution to the “birthday” problem was obvious from near the beginning because it is just like the “boat” problem |
| 11 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I made a correspondence between the parts of the “boat” problem and the parts of the “birthday” problem and then wrote the solution to the “birthday” problem |
| 12 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I did not make much of a link between the town names in the “boat” problem and the names of the people in the “birthday” problem. I just used the same route of the “boat” problem solution to solve the “birthday” problem |
| 13 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I broke the “birthday” problem into smaller pieces and used the same strategies that were used in the “boat” problem solution |
| 14 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I could not solve the “birthday” problem because I got stuck in the same way as I did trying to understand the solution to the “boat” problem |
| 15 | Yes | <input type="radio"/> | No | <input type="radio"/> | O | I did not really use the “boat” problem solution because I already knew the general principle of how to solve problems of this kind |

References

- Carbonell JG (1986) Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. In: Michalski R, Carbonell J, Mitchell T (eds) *Machine learning: an artificial intelligence approach* Los Altos. Morgan Kaufmann, CA
- Catrambone R, Holyoak KJ (1989) Overcoming contextual limitations on problem-solving transfer. *J Exp Psychol Learn Mem Cogn* 15:1147–1156
- Chen Z (2002) A hierarchical analysis of procedural similarity. *J Exp Psychol Learn Mem Cogn* 24:81–95
- Christofides N (1975) *Graph theory: An Algorithmic Approach*. Academic Press, New York
- Duncker K (1945) On problem solving. *Psychol Monogr* 58:1945
- Falkenhainer B, Forbus KD, Gentner D (1989) The structure mapping engine: algorithms and examples. *Artif Intell* 41: 1–63
- Gentner D (1983) Structure mapping: a theoretical framework for analogy. *Cogn Sci* 7:155–170
- Gerjets P, Scheiter K, Kleinbeck S, Schmid U (2002) Learning from transformational and derivational worked-out examples. 24th Conference of the Cognitive Science Society, George Mason University, Washington, 7–11 August, 2002 (member abstract)
- Gick ML, Holyoak KJ (1983) Schema induction and analogical transfer. *Cogn Psychol* 15:1–38
- Holyoak KJ, Thagard P (1989) Analogical mapping by constraint satisfaction. *Cogn Sci* 13:295–355
- Holyoak KJ, Thagard P (1997) The analogical mind. *Am Psychol* 52:35–44
- Hummel J, Holyoak K (1997) Distributed representations of structure: a theory of analogical access and mapping. *Psychol Rev* 104:427–466
- Keene M, Ledgeway T, Duff S (1994) Constraints on analogical mapping: a comparison of three models. *Cogn Sci* 18:387–438
- Novick LR (1988) Analogical transfer, problem similarity, and expertise. *J Exp Psychol Learn Mem Cogn* 14:510–520
- Novick LR (1990) Representational transfer in problem solving. *Psychol Sci* 1:128–132
- Novick LR, Hmelo CE (1994) Transferring symbolic representations across non-isomorphic problems. *J Exp Psychol Learn Mem Cogn* 20:1296–1321
- Ormerod TC, Manktelow KI, Stewart AP, Robson EH (1990) The effects of content and representation on the transfer of Prolog reasoning skills. In: Gilhooly KJ, Keane MT, Logie R, Erdos G (eds) *Lines of thinking: reflections on the psychology of thought*, vol 1. Wiley, Chichester
- Reed SK, Ackinclose CC, Voss AA (1990) Selecting analogous problems: similarity versus inclusiveness. *Mem Cogn* 18:83–98
- Sander E, Richard JF (1997) Analogical transfer as guided by an abstraction process. *J Exp Psychol Learn Mem Cogn* 23:1459–1483
- Schmid U, Carbonell J (1999) Empirical evidence for derivational analogy. In: Wachsmuth I, Jung B (eds) *Proceedings der 4. Fachtagung der Gesellschaft f. Kognitionswissenschaft (28.9.–1.10.99, Bielefeld)* pp 116–121
- Veloso M, Carbonell J (1993) Derivational analogy in PRODIGY: automating case acquisition, storage, and utilization. *Mach Learn* 10:249–278