The Role of Analogy in a Theory of Problem-Solving

Beth Adelson, Mark Burstein, Dedre Gentner, Kristian Hammond, Keith Holyoak, Paul Thagard

The processes that underlie the generation and use of analogies have consistently been of interest in the study of cognition. The goal of this symposium is to look at several computational theories of analogy and to see in what ways each contributes to our understanding. We hope to attain this goal by looking at the sufficiency of each one in the broader context of problem-solving and by asking the following questions:

1. In what way is a theory of analogy constrained by specifying its role in a theory of problem-solving?
2. To what extent are the theories presented below competing theories? To what extent are they members of the same class? Does each one speak to any issue that the others do not address?
3. How is structure related to purpose and semantics?
4. How do purpose and semantics affect:
   (a) Retreival
   (b) Mapping
   (c) Justification
   (d) Debugging
   (e) Generalization

   Present an example reflecting your theory's position.

The Structure-mapping Engine: A Cognitive Simulation of Analogy

Dedre Gentner, University of Illinois, Champagne-Urbana[1]

Computational modeling of analogy.

The Structure-mapping Engine (SME), written by Brian Falkenhainer and Ken Forbus, is a computer simulation of Gentner's structure-mapping theory of analogy (Falkenhainer, Forbus, & Gentner, 1986, in press; Gentner, 1980, 1983, 1988). Given predicate calculus representations of two potential analogs, it uses purely structural principles – ***one-to-one correspondence***, ***structural consistency***, and ***systematicity*** – to interpret and evaluate an analogy between two situations. It operates by first finding all possible relational identities between base and target; it then assigns each of these match hypotheses an evaluation, based on the structural closeness of the match and on a kind of local systematicity by which a given pair of matching predicates is assigned a higher evaluation if their parents also match. SME then sweeps these matching pairs into the largest possible sets consistent with the structural constraints laid out above and computes an overall evaluation. In addition, it hypothesizes *candidate inferences*: new facts about the target domain that are derived by analogy with the base domain. Thus, SME simulates both the *matching* of existing predicates in the two domains and the *carryover* of hypothesized predicates from one domain to the other.

There are some other points to note about the simulation:

1. SME's evaluation concerns only structural soundness. The *validity* of the inferences in the target and the contextual *relevance* of the inferences must be evaluated by separate processes (See Burstein, 1983; Collins & Burstein, in press; Gentner, 1988).

2. To allow us to check our modeling assumptions individually, SME is constructed modularly. For example, different kinds of structural evaluation rules can be tested, and (as discussed below) different predicate-matching rules can be utilized.

[1] Researchers who contributed to this work include Ken Forbus, Brian Falkenhainer, Mary Jo Rattermann, Bob Schumacher, and Janice Skorstad. This research is supported by the Office of Naval Research, Contract No. N00014-85-K-0559.

3. To my knowledge, SME has the greatest range of application of any existing analogy program. Over 40 analogies have been run successfully – that is, they have yielded human-like interpretations and evaluations. Further, SME is efficient. It takes only seconds for most examples.

4. In addition to simulating analogy, SME can also be used to simulate other kinds of similarity: e.g., *mere-appearance* matches, in which only low-order information such as object attributes are considered, and *literal similarity* matches, in which both relational structure and object properties are considered. This enables us to simulate different aspects of human similarity processing.

Psychological studies of access and inference.

Besides analogical mapping, there are other subprocesses in analogical reasoning. Given a current problem (the *target situation*), the person must *access* a similar base situation, *create a mapping* from the base to the target, *draw new inferences* on the basis of the mapping, and *judge* the soundness of the analogy and the relevance and target validity of the candidate inferences. In our recent research we examine the determinants of these subprocesses, using SME to computationally model the results of psychological experiments.

In a series of studies, we gave people different kinds of similarity matches to discover (a) which kinds of matches lead to reminding and (b) which kinds of matches are considered inferentially sound (Gentner & Landers, 1985; Rattermann & Gentner, 1987). Subjects were given roughly 30 short stories to read and remember. A week later, they returned and read a new set of stories; they were to write down any of the original stories that they were reminded of while reading the new stories. The new stories were designed to match the original stories, either as structural analogies or as superficial mere-appearance matches. The results show a dissociation. In rating soundness, subjects rated on the basis of relational commonalities: analogies were rated high and mere-appearance matches low. But their natural remindings showed the opposite pattern: superficial matches were far more likely to be retrieved than relational matches (Holyoak & Koh, 1987; Ross, B. H., 1984). Thus the matches that came most readily to memory were not the matches subjects found inferentially sound.

We have compared the performance of SME with that of our subjects for a subset of the stories (Skorstad, Falkenhainer & Gentner, 1987). We find that the results of the soundness task are best fit by running SME in analogy mode, while the results of the access task are best fit by running SME as a mere-appearance matcher. These results suggest that surface similarity is important in determining access to similarity matches, while relational similarity is important in judging the soundness of a match.

Analogical Problem Solving: A Constraint Satisfaction Approach

Paul Thagard, Cognitive Science Laboratory, Princeton University
Keith Holyoak, Psychology Department, UCLA[2]

We are developing a general cognitive architecture for problem solving and learning in which analogical problem solving will have an important role. We are aiming for a system that will incorporate all the standard components of analogical problem solving in which a source problem is used to help solve a target problem: (1) the retrieval of a potentially useful source, (2) mapping of the target to the source, (3) transfer of the solution of the source to provide a solution to the target, and (4) learning that will facilitate later problem solving, for example by forming schemas that abstract from the source and target.

Our first implementation of analogical problem solving in the PI[3] system was found to put insufficient constraints on the mapping and retrieval processes (Holland, Holyoak, Nisbett, & Thagard, 1986; Holyoak & Thagard, 1986). Accordingly, we have been developing new theories of mapping and retrieval that specify a collection of principled constraints on when an analog will be retrieved and on how components of two analogs will be mapped to each other. Mapping and retrieval both involve structural, semantic, and pragmatic (purpose-directed (Burstein & Adelson, 1987, 1988)) constraints, although the constraints vary in importance to the two processes, with semantics being more important for retrieval than for mapping.

[2]This research is supported by Contract MDA903-86-K-0297 from the Army Research Institute.

[3]Processes of Induction

In mapping, the central constraint is structural correspondence between the two analogs, a relation whose importance has been emphasized by Dedre Gentner. We maintain, however, that semantic correspondences relating predicates with similar meanings are also important. Moreover, pragmatic factors involving the purpose of the analogy can also play a role in mapping. For example, if the purpose of the analogy is to convince someone of a conclusion, then mappings that support this conclusion will be encouraged.

Connectionist models provide a graceful means for *simultaneously* satisfying multiple constraints. Accordingly, we have implemented our theory of mapping in a program called ACME[4] that takes two analogs as inputs and constructs a network of hypotheses concerning what components of the two analogs to map to each other. ACME has now been applied to more than 20 complex analogies drawn from several domains, including radiation problems of the sort investigated experimentally by Holyoak.

Complementary to ACME, we are now developing ARCS [5], a constraint satisfaction model of *retrieval* that uses semantic, structural and pragmatic constraints to help find relevant analogs stored in memory (Holyoak & Thagard, 1987). In contrast to ACME, semantic constraints take precedence in ARCS, with the retrieval of analogs initiated through associations of semantically similar concepts. However, the retrieval process is also guided by structural correspondences and pragmatic import. ARCS is also a connectionist program and is being tested on a large data base.

Eventually, we plan to integrate ARCS and ACME with our rule-based problem solver PI, producing an architecture capable of both analogical and non-analogical problem solving.

Purpose Guided Analogical Learning and Reasoning

Beth Adelson Tufts University, Cambridge, MA[6]

The goal of this research program, conducted jointly with Mark Burstein, is the development of a theory of *purpose-guided* analogical learning and reasoning. Our current work focusses explicitly on the role of partial models in the generation of analogical mappings, and suggests that the process of integrating these multiple analogies which form partial explanations can be described using a set of general principles for relating partial mental models of different types (Burstein & Adelson, 1987).

Our approach of specifying mapping using a principled set of partial models is based on the fact that one typically knows a large amount about a familiar domain and what is mapped from the familiar domain to the domain being learned, is constrained by the *purpose* of the analogy(Burstein & Adelson, 1988; Holyoak & Thagard, 1986). Additionally, although behavior, mechanism, or physical and functional topology may be focussed on during initial learning, full understanding of a complex domain requires the integration of these aspects. In what follows we present some of the issues generated by our theory.

Purpose guided debugging: The role of within domain analogy

Because analogies, by definition, do not provide perfect models of a target domain, a newly mapped model will need to be debugged; structure will need to be refined, added or dropped. Here as in other aspects of analogical learning, purpose can help to constrain the process.

For example, in one of our protocol experiments a student was taught about the concept of stacks by analogy to a stack of plates in a cafeteria. The student was then asked to write the Pascal procedure for pushing elements onto the stack. Using the plate analogy, the student was able to draw correctly a box and arrow representation of the steps involved in push. However, the student was not able to then write the first line of code in which the new element is placed on the stack by pointing the new element's next-pointer to the element that previously was the first element. What needs to be done here is to *assign* to the new element's next-pointer variable the value contained in the head-pointer variable (NEW:NEXT := HEAD). When the student was reminded that pointer variables were analogous to other types of Pascal variables he was able to write the code. That is, he used this within domain analogy to refine his representation of pointer variables.

[4] Analogical Constraint Mapping Engine

[5] Analogical Retrieval by Constraint Satisfaction

[6] This research is supported by the National Science Foundations Engineering Design and knowledge and Data Base Programs

They then took on the properties of other types of Pascal variables; they contained values of a specified type and these values could be copied using the assignment operator.

As this example illustrates, when the learner's task is to describe how a system will be realized in the target domain, base domain analogies will most likely be insufficient. However, within domain analogies may the type of analogies that are appropriate both for this purpose.

Integrating behavioral, causal and topological models

Our theory also addresses the nature of the relationship between behavioral, causal and topological models, how this information is used and why it is important in any problem-solving in which an old solution will be transformed to solve a new, similar problem.

Frequently problem solving will involve understanding the behavior of a system in terms of the relationship between input and output or start and goal states. This description of the system's goal that is contained in a *behavioral* model provides an explanation of the system's purpose. By comparison, a *causal* model represents a system as a connected set of components with causal effects and constraints. The causal model consists of a description of how the outputs of components cause state changes in other, topologically connected components. As a result, a causal description provides an account of what happens across the system in order to get from a start to a goal state. A *topological* model is needed to describe the physical realization of a system's mechanism and behavior. A topological model does this by describing both the *functionality* of the system's components and the interconnections among them.

In our theory, the three models are related in the following way. The behavior model describes the purpose for which the system is used. The causal model is related to the behavioral model in that it describes what is done to acheive the purpose stated in the behavioral model. The topological model explains how the causal model is realized.

The need for the three models and for rules describing their relationship is illustrated by the same student learning about the concept of a queue by analogy to the concept of a stack. The student was told that a queue is like a stack except that it is used when First In First Out behavior is desired. The student was then asked to write the Pascal procedures for pushing and popping stack elements.

The student knew that the Last In First Out behavior of a stack was obtained through a mechanism in which the next element to be popped was the one that was *most* recently pushed. Using the difference in the *behavior* of the two concepts he mapped a transformed *causal* model of a queue, in which FIFO behavior would be obtained by popping the *least* recently pushed element. On the basis of the newly mapped causal model the student was then able to map a transformed *topological* model of a queue. In this model, pushing and popping occur at the same, rather than opposite, ends. Also, as a result of understanding the way in which causal and topological models are related, the student was able to state that the topological model for a queue, as opposed to a stack, contains two pointers rather than one, in order to indicate separately where the next push and pop occur.

Here, and generally in situations in which existing mechanisms are used as analogies in order to implement some new desired functionality, a description of the relationship between the behavioral, the causal and the topological representation enters strongly into successful problem-solving.

Analogical Explanations Combining Inconsistent Mental Models

Mark Burstein BBN Laboratories, Cambridge, MA[7]

Analogies are used during problem solving for a number of purposes. They can be used to suggest plans of action, organizations for components in a synthesis or design problem, predictions of effects of partially understood systems, and explanations of observed behaviors. All of these uses of analogy share some common underlying cognitive mechanisms, but emphasize and exploit semantically different relational structures. We reported a preliminary categorization of some of these types of relational structure in (Burstein & Adelson,

[7]This research was supported in part by the Army Research Institute.

1987).

As described in the preceding section, we are developing a theory of analogical reasoning and learning that differentiates these different uses of analogies. One important aspect of our theory is that analogies can compare and relate systems at several levels of description. Well understood domains or systems, i.e., those whose behavior, mechanism, and applications are all understood, may serve as source analogs at several different levels of abstraction, *and* for several of these different purposes.

One recent computer model that demonstrates some of these uses of analogy is Falkenhainer's PHINEAS system (Falkenhainer, 1987). PHINEAS is an ambitious system that coordinates some large programs for qualitative reasoning and planning, in conjunction with his Structure Mapping Engine (SME) (Falkenhainer, Forbus & Gentner, 1986). Falkenhainer's system first (1) builds a qualitative causal model of heat flow by analogy to liquid flow to *explain an observation* of heat flow *behavior*. The system then seeks to (2) *verify further predictions* from its new model, as a means of testing its reliability and generality. This step also involves (3) *planning experiments* to bring about other situations predictable from the same qualitative model.

One of the reasons the behavior of PHINEAS differs from the model we are developing is that PHINEAS does not use analogical reasoning from its source domain knowledge in steps 2 and 3. That is, its predictions of related target domain behaviors and it's plans for actions to bring about those behaviors are generated using only the newly constructed target domain qualitative model. PHINEAS does not have access to stored plans suggesting uses of water flow for different purposes, nor does it use alternate envisionments of water flow situations to generate additional heat flow predictions. Its resulting model of heat flow is embeded in a specific setting based on the initially presented situation. We would argue that creation of a full test of the generality of the newly derived heat flow *principle*, and the analogy to fluid flow, would involve *extending* the analogy to relate a number of "parallel" water and heat flow situations (Burstein, 1988).

The point of this example is that successful, strategic use of analogies in learning and problem solving is often based on several levels of correspondence between domains. PHINEAS and other systems have successfully shown how the discovery of corresponding *behaviors* can lead to the induction of a corresponding qualitative *causal structure*, but many analogies can be used to relate other levels of description as well. We are developing a computer model where several levels of description can be successively mapped from a single analogy, based on an initial, successful mapping at the behavioral or some other level. This requires richer source domain representations with multiple related examples and mechanisms which allow semantically different kinds of structural relations to be preferred for mapping at different times.

Another reason for investigating this form of purpose-constrained analogical mapping is that human mental models of domains are are seldom complete, nor totally internally consistent (Collins & Gentner, 1987; Spiro et al., 1988; Burstein, 1985). We often use a *set* of related examples with associated explanatory models to keep from generating erroneous predictions and explanations. Similarly, explanations of different behaviors within a single domain may be generated from different analogical models, or combinations of models. Inconsistencies in these explanations can only be detected during problem solving episodes that fully exercise the interrelations between models. We will present examples of subjects developing and relating models derived from different analogies, and show how detected inconsistencies can be understood in terms of attempts to relate mental models at different levels of abstraction during problem solving.

Analogical Reasoning as a By-Product of Problem-Solving.

Kristian J. Hammond, University of Chicago[8]

We *don't* work on analogy. We *do* work on problem solving. As it turns out, we have a program and an approach that results in "analogical" behavior. Simply stated, the program (POLYA) seeks cases in memory to use as exemplars and often finds cases that an observer would see as analogical. POLYA itself,

[8] This work was done with Thomas McDougal at the University of Chicago AI Lab.

however, does not distinguish between true analogy and similarity. POLYA simply uses mapping information associated with the shared predicates to transfer the solution or partial solution of the recalled problem to the new situation without distinguishing between types of similarity.

POLYA is a problem-solver in the domain of geometry theorem proving. In particular, it is a *case-based* problem-solver that constructs new proofs out of existing solutions (or generalizations of solutions) that it finds in memory. The aim of the POLYA project is the development of techniques for automatically constructing vocabularies of feature combination and interaction that are effective in describing problem situations and thus organizing solutions in memory.

The POLYA project is predicated on the notion that for every domain, there exists a vocabulary of feature combination and interaction that best describes the problems within it. The two features that are most important to this vocabulary are the near independence of the predicates and a close association between problem descriptions and solution sets. The near independence of predicates enables use in indexing. The association between problem description and solution set determines utility.

This vocabulary is generated out of the constraints of existing problem/solution pairs and is tested through its use in indexing cases in memory. Those statements used in a proof that, by necessity, share arguments are compounded into a single predicate. A special purpose mechanism is then constructed to recognize this predicate and the predicate is added to the list of features used to index the solution in memory. As new problems are presented, they are understood in terms of the new predicates created by the system and these descriptions are used to search for existing solutions. Rather than confront the general problem of indexing off of arbitrary conjuncts of predicate calculus statements (potentially exponential), only certain conjuncts are recognized and used.

While POLYA constructs these new predicates they do not differ formally from those it initially uses. While a compound predicate such as SQUARE-WITHIN-CIRCLE or KITE-WITH-CROSS represents interactions between parts, the low-level predicates such as LINE and CURVE do as well. Further, the techniques for recognizing the later are of the same type as those for recognizing the former. Rather than look for analogies, POLYA simply looks for useful exemplars.

There are five steps to the POLYA architecture:

1. Proofs are used to generate compound predicates out of conjuncts of predicates implicated in the proof that share variables.
2. New problems are analyzed in terms of these predicates.
3. The resulting predicate list is used to search for appropriate solutions.
4. Mapping information associated with each predicate is used to obtain the bindings between source and target.
5. Any gaps in the proof are dealt with as a new problem description and POLYA recurs on the search for a solution.
6. The resulting proof is used to generate a new set of candidate predicates.

The result of this is a system that makes use of cases that are both "similar" and "analogous" to its problem situation without having to distinguish between the two. The behavior is analogical, but this is a natural by-product of the case-based problem solving.

References

Burstein, M. H. (1983). A model of learning by analogical reasoning and debugging. In *Proceedings of the International Conference on Artificial Intelligence*, Washington, D. C.

Burstein, Mark H. (1985). Learning by Reasoning from Multiple Analogies. Doctoral dissertation, Yale University.

Burstein, Mark H. (1988) Incremental Learning from Multiple Analogies. *ANALOGICA: The first workshop on analogical reasoning.* Los Altos, CA: Morgan Kaufmann.

Burstein, M. H. and Adelson, B. (1987) Analogical Learning: Mapping and Integrating Partial Mental Models. In *Proceedings of the 1987 Conference of the Cognitive Science Society*. University of Washington, Seattle, WA.

Burstein, M. and Adelson, B. (1988) Analogical Reasoning for Learning. in *Applications of Artificial Intelligence to Educational Testing*. R. Freedle (Ed.) In press. Erlbaum: Hillsdale, NJ.

Collins, Allan and Gentner, Dedre. (1987) How People Construct Mental Models. In N. Quinn and D. Holland (Eds.), *Cultural Models in Thought and Language*. Cambridge, UK: Cambridge University Press.

Collins, A., & Burstein, M. (in press). A framework for a theory of mapping. To appear in S. Vosniadou and A. Ortony (Eds.)*Similarity and analogical reasoning*.

Falkenhainer, B., Forbus, K. D. & Gentner, D. (1986). *The structure-mapping engine Proceedings of the Meeting of the American Association for Artificial Intelligence*. Philadelphia. Revised version to appear in *Artificial Intelligence*, in press.

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

Gentner, D. (1988). Analogical inference and analogical access. In A. Prieditis (Ed.), *Analogica*. Los Altos, CA: Morgan Kaufmann.

Gentner, D., & Landers, R. (1985). Analogical reminding: A good match is hard to find. In *Proceedings of the International Conference on Systems, Man and Cybernetics* (pp.607-613). Tucson, AZ.

Holland, J., Holyoak, K., Nisbett, R., and Thagard, P. (1986). Induction: Processes of inference, learning, and discovery. Cambridge, MA: Bradford Books/MIT Press.

Holyoak, K. J., & Koh, K. (1987). Surface & structural similarity in analogical transfer. *Memory and Cognition*, *15*, 332-340.

Holyoak, K. and Thagard, P. (1986). A computational model of analogical problem solving. To appear in S. Vosniadou and A. Ortony (eds.), Analogy, Similarity, and Thought. (Cambridge: Cambridge University Press, in press.)

Holyoak,K. and Thagard, P. (1987). Analogical mapping by constraint satisfaction: A computational theory. Unpublished manuscript.

Ross, B. H. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology*, *16*, 371-416.

Skorstad, J., Falkenhainer, B., & Gentner, D. (1987, August). Analogical processing: A simulation and empirical corroboration. In *Proceedings of Meeting of the American Association for Artificial Intelligence*, Seattle, WA.

Falkenhainer, Brian. (1987) Scientific Theory Formation Through Analogical Inference. In *Proceedings of the Fourth International Workshop on Machine Learning*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.

Spiro, R.J., Feltovich, P.J., Coulson, R.L., Anderson, D. (In Press) Multiple Analogies for Complex Concepts: Antidotes for Analogy-Induced Misconception in Advanced Knowledge Acquisition. In S. Vosniadou and A. Ortony (Eds.), *Similarity and Analogical Reasoning*. New York, NY: Cambridge University Press.