

# Strategy Variations in Analogical Problem Solving

Tom Y. Ouyang and Kenneth D. Forbus

Qualitative Reasoning Group, Northwestern University  
2133 Sheridan Road  
Evanston IL, 60208  
ouyang@csail.mit.edu, forbus@northwestern.edu

## Abstract

While it is commonly agreed that analogy is useful in human problem solving, exactly how analogy can and should be used remains an intriguing problem. VanLehn (1998) for instance argues that there are differences in how novices and experts use analogy, but the VanLehn and Jones (1993) Cascade model does not implement these differences. This paper analyzes several variations in strategies for using analogy to explore possible sources of novice/expert differences. We describe a series of ablation experiments on an expert model to examine the effects of strategy variations in using analogy in problem solving. We provide evidence that failing to use qualitative reasoning when encoding problems, being careless in validating analogical inferences, and not using multiple retrievals can degrade the efficiency of problem-solving.

## Introduction

Analogy has long been viewed as one of the central methods for learning and discovery (cf. Polya, 1954; Carbonell, 1983), and AI researchers have found that analogy can provide significant computational benefits (cf. Veloso 1994; Melis & Whittle, 1999). Yet VanLehn and Jones (1993) argued, based on protocol analyses of human students, "There is also evidence that good solvers avoid analogical problem solving in general.", and argues that "principle-based reasoning should be used instead of analogy during the equation generation step [of problem solving]." How can we resolve these conflicting views?

One resolution was suggested by Gentner *et al.* (1997), who argued that, broadly speaking, analogy can be used in two very different ways by learners. First, analogy can be used as a form of X-ray, comparing two situations to reveal common relational structure that provides new insights. Second, analogy can be used as a kind of recipe, to avoid thinking. Consider filling out tax forms: Most of us do this not by thinking through the principles underlying the tax code, but by going over last year's form and doing the same thing, only with this year's numbers. For students trying to solve a problem easily, analogy provides a way to minimize thinking. For students trying to solve the problem to learn, analogy provides a way to garner new insights.

Subsequently, more detailed analyses of protocol data by VanLehn (1998) caused him to revise their earlier theory, arguing instead that both novices and experts used analogy, but that their strategies for using analogy were different. This raises a number of important theoretical questions. What strategies lead to effective and accurate problem solving? What strategies make the most sense for learners to use? Here we focus on the first question, determining what aspects of analogical problem-solving strategies lead to poor performance and results and what aspects lead to more expert performance.

We investigate this question by using a system based on Pisan's (1996; 1998) *Thermodynamics Problem Solver* (TPS). This system, the *Analogical Problem Solving System* (APSS), extends TPS with the analogy ontology developed in (Forbus *et al.* 2002). The addition of declarative strategies for using analogy in problem solving allows us to easily change and "ablate" the system's capabilities. We show by ablation studies how different ways of using analogy can lead to particular improvements or degradations in performance. In particular, we examine how the encoding of the problem, careful validation of actions suggested by analogies, and the number of retrievals for each problem affect both the accuracy of solutions and how much effort was required to produce them. These analyses provide evidence for the claim that variations in analogical problem solving strategies can indeed explain differences between novice and expert performance.

We start by briefly reviewing the analogy theory and simulations used in this work. Next, we describe the architecture of the system, followed by the experiments. Finally, we discuss related and future work.

## Background

There is now considerable empirical support for Gentner's (1983) structure-mapping theory as a model of human analogy and similarity (e.g., Markman & Gentner, 2000). Structure-mapping theory defines analogy and similarity in terms of comparisons involving structured descriptions. Comparisons start with a *base* description and a *target* description, and result in one or more *mappings*. A mapping consists of *correspondences*, *candidate inferences*, and a *structural evaluation score*. A *correspondence* aligns entities or statements of the base

```

(defmodelfragment IdealGasLaw
  :participants
  ((Stuff :type ThermodynamicStuff
    :constraints
    ((substanceOf Stuff Sub))
    (Sub :type Substance))
  :conditions ((idealGas Stuff))
  :consequences
  ((mfEquation IdealGasLaw
    (= (* (P Stuff) (V Stuff))
      (* (Mass Stuff) (R Sub) (T Stuff))))))

```

Figure 1: A compositional modeling fragment that describes the ideal gas law

with those of the target. *Candidate inferences* represent conjectures about the target made by projecting non-overlapping base structure onto it. Finally, the *structural evaluation score* indicates the overall match quality.

We use two simulations built on structure-mapping theory in this work. The Structure-Mapping Engine (SME) (Falkenhainer *et al.* 1989) is a model of analogical matching. It takes as input the base and target representations, and computes mappings according to the principles outlined above. SME is efficient, using a greedy algorithm to construct mappings in polynomial time, and can respond to incrementally available data (Forbus *et al.* 1994a). MAC/FAC (Forbus *et al.* 1994b) models similarity-based retrieval in two stages. The MAC stage exploits a special kind of feature vector, automatically constructed from structured representations, called *content vectors* to find library cases that are similar to the probe case. The dot product of two content vectors provides an estimate of the size of the mapping SME might compute for the structured representations. The best cases are passed to the second stage, FAC, which uses SME in parallel to match the structured descriptions for the items retrieved by the MAC stage. The best match, plus any that score within 10% of it (up to three in all), are returned as reminders.

SME and MAC/FAC have been used to successfully model a variety of psychological phenomena, and SME has been used to make new predictions which have subsequently been confirmed via psychology experiments. Both have also been used in successful performance systems previously (Forbus, 2001 summarizes). Thus these simulations provide a good starting point for modeling analogical problem solving.

## Domain Modeling

The domain theory used in our system is that from Pisan's (1998) TPS. It uses the compositional modeling framework (Falkenhainer and Forbus, 1991) to represent the physical and mathematical knowledge needed to solve textbook thermodynamics problems. This framework consists of three types of components: modeling fragments, entities, and relations. Model fragments describe some fundamental piece of the domain's physics, such as *conservation of mass* or *pressure equilibrium in a*

*container*. Entities represent types of physical objects in the domain such as *pistons* or *cylinders*. Relations describe logical relations such as *inside* or *inletOf*.

A model fragment becomes active when its participants are present in the scenario and all of its conditions are satisfied. For example, if the problem describes ThermodynamicStuff *S1* as an *ideal gas*, the system would activate an instance of the model fragment *IdealGasLaw*, shown in Figure 1. Multiple instances of a model fragment can be active for different sets of participants, so another instantiation of *IdealGasLaw* could apply to ThermodynamicStuff *S2* in the same problem. By activating a model fragment, the system also activates its consequences. In this case, the consequence is the equation stating the ideal gas law, which becomes available to the problem solver. Through this activation process, the model fragments in the domain theory combine with the statements in the problem description to form a complete problem scenario.

## Problem Solving

### First Principles Reasoning

Like TPS, APSS's reasoning system uses the suggestion architecture proposed in (Forbus & de Kleer, 1993). Each suggestion represents a plan for achieving a goal, by satisfying an ordered list of subgoals. Carrying out a suggestion can either solve the parent goal directly or introduce new subgoals that must be satisfied. This process continues until the original problem is solved, the system runs out of suggestions to try, or until resource bounds are exceeded.

```

Suggestion: SolveEquation
Form: (SolveEquation ?eqn ?quantity ?val)
Goal: (value ?quantity ?val)

Subgoals:
1: Find a valid ?eqn that mentions ?quantity
2: For all ?unknown in ?eqn
   Solve (value ?unknown ?ukval)
   Bind ?unknown in ?eqn with ?ukval
3: Solve for ?quantity in bounded ?eqn

```

Figure 2: An APSS suggestion that finds a numerical quantity by solving an equation.

In the APSS system, one common goal is to find the value for some unknown quantity. One way to achieve this goal is to search for an equation that mentions the quantity in question. The suggestion for performing this, shown in Figure 2, defines a list of ordered subgoals to (1) search for valid equations that mention the quantity in question, (2) extract and solve for all unknowns in the equation and replace them with their derived numerical values and (3) solve the equation for the given quantity once all unknowns have been replaced. In this example, only *?quantity* is bound initially. The variable *?eqn* in

is not mentioned in the goal form, so its binding is not passed to the parent goal.

## Analogical Reasoning

Analogical reasoning allows APSS to reuse past experience to solve new problems more effectively. Instead of starting from scratch, it can take advantage of the successes and failures from previous problems to guide the search for a solution. The analogical problem solving model is divided into two stages - retrieving examples that are similar to the proposed problem and transferring knowledge from the past case to the new problem.

When APSS starts solving a new problem, it first generates a description of the problem scenario that serves as the probe for similarity based retrieval. The probe is then compared with every example in the case library using MAC/FAC, which returns up to three reminders that represent the best estimated matches. (The exact content of the probe and the number of retrievals also depends on the analogy strategy, which we explain in more detail in the Experiments section).

After retrieving similar examples from the case library, the system uses SME to generate a mapping from each of the base cases (the retrieved examples) to the target case (the new problem). Recall that correspondences align common entities and statements between the base and target, and candidate inferences transfer statements that from the base that might hold in the target. Since the base case includes the solution trace, the resulting candidate inferences effectively translates the solution in the base case to the problem described in the target case.

Like first principles reasoning, analogical reasoning is implemented using the suggestion architecture. The suggestion *FindNvalueUsingAnalogy*, illustrated in Figure 3, first checks whether there is a matching goal in any of the previous example by searching the set of candidate inferences. For each potential match it verifies that the inferred suggestion is valid for the current problem by attempting to re-derive it from first principles knowledge. Finally, the suggestion (3) attempts to solve the goal by performing the inferred suggestion.

## Integrating Analogy and First Principles

Since it is usually not possible to derive a solution using only analogy, APSS uses both analogical reasoning and first principles reasoning and switches from one to the other as needed at any stage in the problem solving process. Using this system, we analyze the hypothesis that suggestions generated by analogy are usually more reliable than those that are derived only through first principles. In cases where both types of suggestions exist, the system

```
Suggestion: FindNvalueUsingAnalogy
Goal: (value ?quantity ?val)

Subgoals:
1: Retrieve candidate inferences that match:
   (implies ?step (value ?quantity ?val))
2: Check if ?step is valid
3: Perform ?step to solve for ?val

Match: S1 (Base) ~ S1 (Target)
      C1 (Base) ~ P1 (Target)

Base Case:
(implies
 (solveEquation (= (V S1) (V C1)) (V S1) 2)
 (value (V S1) 2))

Candidate Inference (CI):
(implies
 (solveEquation (= (V S1) (V P1)) (V S1) ?v)
 (value (V S1) ?v))
```

Figure 3: An APSS suggestion (above) that finds a numerical quantity by following steps inferred by analogy and (below) an example of the candidate inferences used to generate the analogy subgoals.

favors the ones backed by analogy and explores those first. This preference for analogy suggestions also extends to the children of the suggestion node, allowing the system to search deeper down branches supported by analogy before backtracking to consider other alternatives.

## Example

Figure 4 shows one of the textbook thermodynamics problems in our corpus, both in its original form and translated to a set of initial facts for the problem solver. For the same problem, Figure 5 shows a partial tree of the solution generated by our system. The root node, *Mass P*, represents the original goal. To solve this goal, the system generates suggestions to perform a table lookup (left child) and to search for an equation that mentions the goal (right child). No valid table lookup exists for *Mass P* so the first suggestion fails. The system then proceeds to the Equation  $Mass = Weight * Gravity$ , generating subgoals for each of the unknown quantities *Gravity* and *Weight*.

```
Problem 2.17: A frictionless piston-cylinder device has
a cross-sectional area of 15 in2. Find the piston mass
such that the absolute pressure in the cylinder is 150
psi.

(isa S ThermodynamicStuff)
(isa P Piston)
(directionOf P Up)
(inside S P)
(nvalue (Acc Gravity) 32.2 ft/s2)
(nvalue (Pressure S) 150 psi)
(nvalue (Area (top P)) 15 in2)
```

Figure 4: An example thermodynamics textbook problem. The original description is given above and the translated facts used by APSS are listed below.

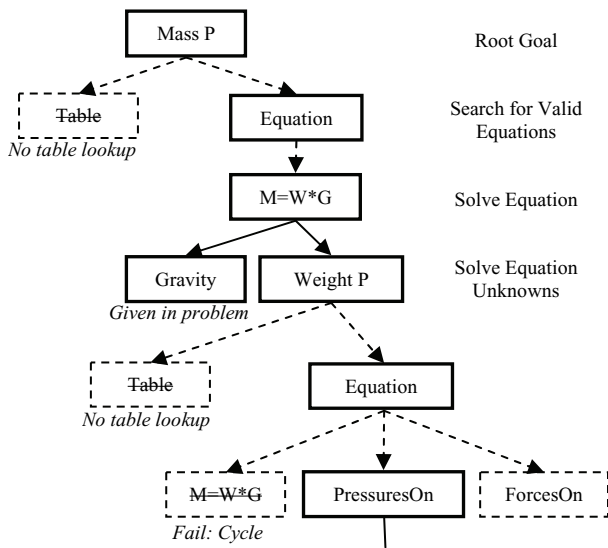


Figure 5: A partial solution trace (and/or tree) for the thermodynamics problem in Figure 4. The solid boxes represent solved nodes and boxes with strikethrough labels represent failed nodes. The solid arrows represent AND links and broken arrows represent OR links.

## Experiments

The goal of our experiments is to determine how different strategies for using analogy affect performance in problem solving. It is well-known that experts are more efficient and more accurate at solving problems in their domain than are novices. Our hypotheses about the role of analogy in these differences are as follows: (1) Experts use analogy whenever possible in problem solving. (2) Experts have better encoding processes than novices, annotating problems with information that makes good retrievals more likely (Chi *et al.* 1981). (3) Experts generate and use multiple reminders more often than novices. (4) Experts carefully check the validity of candidate inferences before relying on their results.

We measure performance by the accuracy of the result and the number of major suggestions (e.g., solving and equation or performing a table lookup) that were expanded. In each of the ablation experiments, we disable aspects of our notional expert model for using analogy in problem-solving to see what their effects are.

The set of test problems used in our experiments consists of 82 textbook problems from the first four chapters of “Fundamentals of Engineering Thermodynamics” by John R. Howell and Richard O. Buckius. For the following experiments, we solve each of the 82 problems with the remaining 81 problems from this corpus acting as the case library.

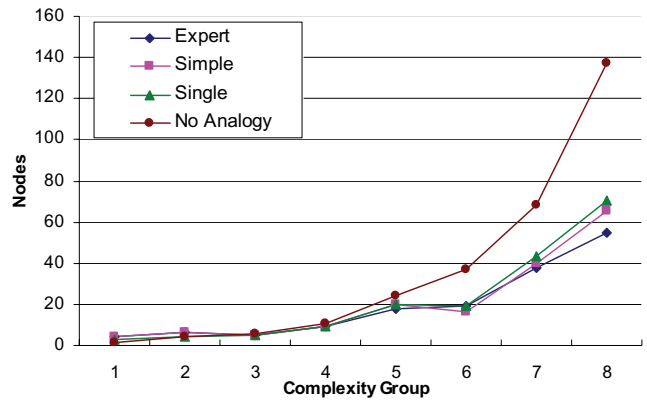


Figure 6: The performance of the different strategies for analogical problem solving on the test problems, grouped into eight sets by complexity as measured by the number of nodes expanded by the non-analogical problem solver.

	Performance	P-Value
<b>Expert</b>	20.2 nodes (0 errors)	
<b>Simple Encoding</b>	21.9 nodes (0 errors)	> Expert p = .083 (.059)
<b>Single Reminding</b>	23.0 nodes (0 errors)	> Expert: p = .051 (.038)
<b>Simple/Single</b>	21.8 nodes (0 errors)	> Expert: p = .095 (.037)
<b>No Validation</b>	19.8 nodes (2 errors)	< Expert: p = .40 (.25)

Table 1: Ablation experiment results showing the average performance and the number of problems solved incorrectly.

Table 1 summarizes the difference that these strategic parameters make, holding all other parameters consistent with the Expert model (i.e., the *Simple Encoding* strategy still uses *Validation* and *Multiple Reminders*). The table also shows the statistical significance of the difference between each strategy and the *Expert* solver (> denotes more nodes expanded). When considering the entire set of 82 problems, only the performance of the *Single Reminding* strategy seems to be significantly different from that of the *Expert* solver. However, when we look at only the most complex quartile of the problems in the corpus, all of the performance differences become more significant (P-values given in parenthesis). This relationship between performance variation and problem complexity is also reflected in the graph from Figure 6.

## Does Analogy Improve Efficiency?

Figure 6 illustrates that analogy does improve efficiency. In this experiment, problems were evenly divided into 8 groups of increasing complexity, as measured by the number of nodes required to solve the problem without using analogy. While the relative performance for individual problems within each group varied, on average the benefit from using analogy grows as the problem complexity increases. This is consistent with the pattern of results found in the literature (cf. Veloso 1994).

## Effect of Rigorous Problem Encoding

One way that experts differ from novices is how they encode problems (Chi *et al.* 1981). To examine this phenomenon, the first ablation experiment compares the performance between two models for problem encoding. The first model only stores the facts given explicitly in the textbook problem while the second model expands this description to include all of the model fragments, relationships, and processes that could be derived through the thermodynamics domain knowledge.

The first two rows in Table 1 show that on average, rigorous problem encoding leads to moderately (9%) better performance compared to simple encoding. However, it is interesting to note that only 9 of the 82 problems account over 95% of the performance difference between the two models. These results, along with the graph in Figure 6, show that rigorous encoding leads to a noticeable improvement in problem solving effectiveness, especially for the more complex problems in the corpus.

One explanation for the difference is that the rigorous encoding enables the retrieval mechanism to focus on the underlying similarities in the thermodynamics concepts instead of on how the problems are phrased in the text. A problem that describes a container with a 490 Newton force pointing down would not contain the same statements as another problem that just mentions a piston under compression. However, conceptually the two scenarios are very similar. Conversely, the simple encoding model overlooked this useful reminding in favor of cases with more surface similarities.

## Effect of Single or Multiple Retrievals

Another factor that affects analogical problem solving is the number of past examples to consider when solving a new problem. The first retrieval is not always the best, and allowing multiple reminders both increases the chance of finding a relevant case and also enables the solver to combine suggestions from multiple examples. In our experiments, the expert model uses all of the MAC/FAC reminders (up to three) retrieved from the case library. The single reminding strategy, in contrast, uses only the highest scoring reminding. We compare the relative performance of these two strategies.

The third row in Table 1 shows that when using rigorous encoding, limiting the number of retrievals noticeably (14%) degrades analogical performance. However, this is no longer the case when we repeat the experiment with both single retrieval and simple encoding, as shown in row four. The strategies for multiple and single reminders lead to nearly identical results when only simple encoding is used. This suggests that multiple retrievals are most useful when combined with rigorous encoding, enabling the system to choose more reliably between the less obvious secondary and tertiary candidates.

## Validation of Analogy Suggestions

In order to save time and effort, people do not always verify that the steps suggested by the retrieved case are valid for the new problem. To model this behavior, we modify the expert strategy so that it applies analogy steps without rigorously re-deriving them through first principles knowledge. The results from the validation ablation experiment show that two problems were solved incorrectly when combining the no-validation strategy with rigorous encoding. However, the performance gain from disabling validation was marginal (about 2%).

In fact, if we exclude the two problems which the no-validation strategy missed, the performance is actually slightly worse than that of the expert model. One reason is that invalid suggestions inferred by analogy often mention quantities that could not be derived from the problem scenario. Exploring these dead-end paths caused extra node expansions, adversely affecting performance. These results seem to support the hypothesis that expert solvers generally do not skip the validation step since failure to do so not only introduces the risk of deriving incorrect answers but also fails to improve performance due to the extra time spent exploring dead ends.

## Related Work

There are a number of competing simulations of analogical matching, some of which also incorporate retrieval and representation construction. The connectionist models (cf. Eliasmith & Thagard, 2001; Hummel & Holyoak 1997; Larkey & Love, 2003), which focus on neural plausibility, cannot handle representations of the size used in this project. Other simulations that focus on representation construction (cf. Mitchell 1993; French 1995) use domain-specific matching algorithms, unlike our use of general-purpose, domain-independent cognitive simulations. Likewise, (VanLehn & Jones, 1993) used hand tailored matchers and retrieval mechanisms.

Most research on analogical problem solving in AI occurs as case-based reasoning (cf. Kolodner, 1994; Leake, 1996). Many current CBR systems use feature vectors, which are inadequate for problem solving of this type, and CBR systems optimize retrieval and matching algorithms for task performance, whereas our matching and retrieval algorithms are designed as simulations of

human analogical matching and similarity-based retrieval. The closest CBR work is PRODIGY-ANALOGY (Veloso & Carbonell, 1993), which was the first broad-scale use of analogy in a cognitive architecture. Methods for identifying representations to improve matching (e.g., Finlayson & Winston, 2003) could provide another dimension to explore regarding variations in expertise.

## Discussion

These experiments provide evidence that a combination of rigorous encoding, analogy suggestion validation, and multiple retrievals resulted in more expert-like performance, and that ablating aspects of these strategies degraded performance, in ways that made the system behave more like novice problem solvers. This is consistent with the protocol evidence analyzed by VanLehn (1998). Since VanLehn's studies have been in college physics and our experiments have used engineering thermodynamics, this lends some support to the theory that these strategies and novice/expert variations could be general across a variety of domains.

We plan to follow up on this work in two ways. First, are the same strategy variations found in everyday reasoning? Studying comprehension questions could shed light on this issue. Second, we plan to incorporate these strategies into the Companion Cognitive Systems architecture (Forbus & Hinrichs, 2004), to experiment with analogical problem solving in the context of an ongoing learning system.

## Acknowledgements

This research was supported by the Cognitive Science Program of the Office of Naval Research, and by the Information Processing Technology Office of DARPA.

## References

- Carbonell, J. 1986. Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R. Michalski, J. Carbonell, and T. Mitchell (eds.) *Machine Learning: An artificial Intelligence Approach*, 137-161.
- Chi, M. T. H., Feltovich, P. J., & Glaser, R. 1981. Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, 121-152.
- Eliasmith, C. and Thagard, P. 2001. Integrating structure and meaning: A distributed model of connectionist mapping. *Cognitive Science*.
- Falkenhainer, B. and Forbus, K. 1991. Compositional Modeling: Finding the Right Model for the Job, *Artificial Intelligence*, 51(1-3), 95-143.
- Falkenhainer, B., Forbus, K. and Gentner, D. 1986. The Structure-Mapping Engine. *Proceedings of AAAI '86*.
- Forbus, K. 2001. Exploring analogy in the large. In Gentner, D., Holyoak, K., and Kokinov, B. (eds.) *The Analogical Mind: Perspectives from Cognitive Science*.
- Forbus, K., Gentner, D., and Law, K. 1994. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19, 141-205.
- Forbus, K. and Hinrichs, T. 2004. Companion Cognitive Systems: A step towards human-level AI. *Proceedings of the AAAI Fall Symposium on Achieving Human-level Intelligence through Integrated Systems and Research*.
- Forbus, K., Mostek, T., and Ferguson, R. 2002. An analogy ontology for integrating analogical processing and first-principles reasoning. *Proceedings of IAAI '02*.
- Finlayson, M. and Winston, P. 2004. A model of analogical retrieval using intermediate futures. *Proceedings of CogSci '04*.
- French, R. 1995. *The subtlety of similarity*. The MIT Press, Cambridge, MA.
- Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy, *Cognitive Science*, 7(2), 155-170.
- Gentner, D., Brem, S., Ferguson, R. W., Markman, A. B., Levidow, B. B., Wolff, P., & Forbus, K. D. 1997. Analogical reasoning and conceptual change: A case study of Johannes Kepler. *The Journal of the Learning Sciences*, 6(1), 3-40.
- Hummel, J.E., & Holyoak, K.J. 1997. Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104 (3), 427-466.
- Kolodner, J. L. 1994. *Case-based reasoning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Larkey, L. and Love, B. 2003. CAB: Connectionist Analogy Builder. *Cognitive Science*, 27, 781-794.
- Leake, D. (ed.) 1996. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. The MIT Press, Cambridge, MA.
- Markman, A. B., & Gentner, D. 2000. Structure-mapping in the comparison process. *American Journal of Psychology*, 113(4), 501-538.
- Melis, E. and Whittle, J. 1999. Analogy in inductive theorem proving. *Journal of Automated Reasoning*, 22, 117-147.
- Mitchell, M. 1993. *Analogy-making as perception: A computer model*. The MIT Press, Cambridge, MA.
- Mostek, T., Forbus, K. and Meverden, C. 2000. Dynamic case creation and expansion for analogical reasoning. *Proceedings of AAAI '00*.
- Pisan, Y. 1996. Using qualitative representations in controlling engineering problem solving. *Proceedings of QR '96*.
- Pisan, Y. 1998. An integrated architecture for engineering problem solving. Doctoral dissertation, Northwestern University, Evanston, IL, USA.
- Polya, G. 1954. *Mathematics and Plausible Reasoning*. Princeton University Press, Princeton, NJ.
- VanLehn, K. 1998. Analogy Events: How Examples are Used During Problem Solving. *Cognitive Science*, 22(19), 347-388.
- VanLehn, K. and Jones, R. 1993. Better learners use analogical problem solving sparingly. *Proceedings of the 10<sup>th</sup> International Conference on Machine Learning*
- Veloso, M. and Carbonell, J. 1993. Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning*, 10, 249-278.
- Veloso, M. 1994. Flexible Strategy Learning: Analogical replay of problem-solving episodes. *Proceedings of AAAI '94*.