Measuring the Level of Transfer Learning by an AP Physics Problem-solver

Matthew Klenk and Ken Forbus
Qualitative Reasoning Group, Northwestern University
2133 Sheridan Road, Evanston, IL 60208 USA
Contact: m-klenk@northwestern.edu

Abstract

Transfer learning is the ability of an agent to apply knowledge learned in previous tasks to new problems or domains. We approach this problem by focusing on model formulation, i.e., how to move from the unruly, broad set of concepts used in everyday life to a concise, formal vocabulary of abstractions that can be used effectively for problem solving. This paper describes how the Companions cognitive architecture uses analogical model formulation to learn to solve AP Physics problems. Our system starts with some basic mathematical skills, a broad common sense ontology, and some qualitative mechanics, but no equations. Our system uses worked solutions to learn how to use equations and modeling assumptions to solve AP Physics problems. We show that this process of analogical model formulation can facilitate learning over a range of types of transfer, in an experiment administered by the Educational Testing Service.

Introduction

The observation that people improve in their ability to learn new skills based upon previous related tasks motivates AI transfer learning research. The task of model formulation, with its emphasis on abstraction and problem solving, is an interesting problem for transfer learning systems. An important contribution of qualitative reasoning has been formalizing the process of model formulation (Falkenhainer & Forbus 1991; Nayak 1994, Rickel & Porter 1994). Most model formulation work has focused on deriving what levels of detail and which perspectives should be used in a model, given a particular task. They take as input a structural description, an abstract high-level schematic, of the system to be modeled. They generally do very little reasoning about everyday concepts and entities, an exception being Flores & Cerda's (2000) work on analog electronics.

One area where this issue arises is solving textbook physics problems. Two model formulation challenges students face in this task are (1) learning the conditions under which particular equations are applicable and (2) learning how to translate particular real-world conditions into parameter values (i.e. an object at the top of a projectile motion event has zero vertical velocity). The AP Physics exam is administered in the US by the Educational Testing Service (ETS) to test high school students' physics problem solving competency. Figure 1 shows four example problems generated by the ETS for the purpose of testing our system. The templates which generated these problems represent roughly 20% of the typical Mechanics portion of the exam.

The motivating hypothesis of the Companions cognitive architecture (Forbus & Hinrichs 2006) is that the flexibility and breadth of human common sense reasoning arises from analogical reasoning and learning from experience. That is, they use their experience (both everyday and with solving textbook problems) to enable them to solve new problems, and over time, extract generalizations and heuristics. This is consistent with Falkenhainer's (1992) observation that engineers often use analogies with their experience to formulate new models. Klenk *et al.* (2005) showed that a Companion can formulate models by analogy to solve everyday physical reasoning problems, of the kind used in the Bennett Mechanical Comprehension test. This paper goes beyond that result by demonstrating

1. A ball is released from rest from the top of a 200 m tall building on Earth and falls to the ground. If air resistance is negligible, which of the following is most nearly equal to the distance the ball falls during the first 4 s after it is released? (a) 20m; (b) 40m; (c) 80m; (d) 160m.
2. An astronaut on a planet with no atmosphere throws a ball upward from near ground level with an initial speed of 4.0 m/s. If the ball rises to a maximum height of 5.0 m, what is the acceleration due to gravity on this planet? (a) 0.8m/s^2; (b) 1.2m/s^2; (c) 1.6m/s^2; (d) 20m/s^2;
3. A box of mass 8kg is at rest on the floor when it is pulled vertically upward by a cord attached to the object. If the tension in the cord is 104N, which of the following describes the motion, if any, of the box? (a) It does not move; (b) It moves upward with constant velocity; (c) It moves upward with increasing velocity but constant acceleration; (d) It moves upward with increasing velocity and increasing acceleration.
4. A block of mass M is released from rest at the top of an inclined plane, which has length L and makes an angle q with the horizontal. Although there is friction between the block and the plane, the block slides with increasing speed. If the block has speed v when it reaches the bottom of the plane, what is the magnitude of the frictional force on the block as it slides? (a) $f = Mg\sin(q)$; (b) $f = Mg\cos(q)$; (c) $f = MgL\sin(q) - ½Mv^2$;(d) $f = [MgL\sin(q) - ½Mv^2]/2$.

Figure 1: Example AP Physics problems

that such techniques can be used to learn to solve systematic variations of AP Physics problems.

An important question in learning is how well what is learned transfers to solving new problems. Here we explore six distinct kinds of near-transfer problems[1]:

1. *Parameterization*: Changing the parameter values, but not qualitative outcome
2. *Extrapolation*: Changing the parameters such that the qualitative outcome changes as well
3. *Restructuring*: Asking for a different parameter
4. *Extending*: Including distracting information
5. *Restyling*: Changing the types of everyday objects involved
6. *Composing*: Requiring concepts from multiple problems

This paper describes how the Companions cognitive architecture (Forbus & Hinrichs 2006) uses analogical model formulation to solve AP Physics problems, including handling these kinds of transfer. We start by briefly reviewing the key aspects of the architecture. The analogical problem solving method, which learns by accumulating worked solutions, is described next. We discuss an experiment, administered by ETS, which shows a Companion is capable of performing these types of transfer. We close with a discussion of related work and our plans to build upon these results.

The Companions Architecture

The Companions architecture is exploring the hypothesis that structure-mapping operations (Gentner 1983; Forbus & Gentner 1997) are central to human reasoning and learning. Forbus & Hinrichs (2006) provides an overview of the Companions architecture; for this paper, the key processes to understand are analogical matching and retrieval. We summarize each in turn and explain how they facilitate transfer.

The Structure-Mapping Engine (SME) models the structure-mapping process of comparison (Falkenhainer *et al.* 1989). Structure-mapping postulates that this process operates over two structured representations (the *base* and *target*), and produces one or more *mappings*, each representing a construal of what items (entities, expressions) in the base go with what items in the target. This construal is represented by a set of *correspondences*. Mappings also include a *score* indicating the strength of the match, and *candidate inferences* which are conjectures about the target using expressions from the base which, while unmapped in their entirety, have subcomponents that participate in the mapping's correspondences. SME operates in polynomial time, using a greedy algorithm (Forbus *et al.*1994a; Forbus & Oblinger 1990).

MAC/FAC (Forbus *et al.* 1994b) models similarity-based retrieval given a case of facts, or *probe,* and a large case library. The first stage, using a special kind of feature vector automatically computed from structural descriptions, rapidly selects a few (typically three) candidates from the case library. The second stage uses SME to compare these candidates to the probe, resulting in one (or more, if they are very close) reminding. Both SME and MAC/FAC have been used successfully in many domains, and as cognitive models, both have been used to model a number of psychological results (Forbus 2001).

These domain independent systems facilitate transferring knowledge at each of the six transfer levels. Because SME and MAC/FAC focus on structural matches, they are insensitive to particular numerical values, easing parameterization transfer. The emphasis on relational structure aids extrapolation and restructuring problems because contextual information in the base remains associated in the candidate inferences. SME and MAC/FAC's ability to handle partial matches facilitates extending and restyling problems. Composing, as explained below, is achieved by multiple retrievals.

Solving Problems by Worked Solutions

When students study for the AP Physics exam, one important way in which they learn is by doing problem sets. For feedback, students often get worked solutions. These step-by-step explanations are frequently found in the back of textbooks. Our system learns by using such worked solutions. In collaboration with ETS and Cycorp, we developed representation conventions for problems and worked solutions. ETS then generated examples from templates (not available to us) for development and testing purposes. The representations used the ontology of the ResearchCyc knowledge base, plus our own extensions. ResearchCyc is useful for this purpose because it includes over 30,000 distinct types of entities, over 8,000 relationships and functions, and 1.2 million facts constraining them. Thus, everyday concepts like "astronaut" and "ball" are already defined for us, rather than us generating them for the purpose of this project.

Example Problem and Worked Solution

Figure 2 shows some of the 37 facts used to represent Problem 2 from Figure 1. The worked solutions are predicate calculus representations of what is found in textbooks. They are not deductive proofs, nor problem solving traces in the language of our solver. This is important, because it provides more opportunities for Companions to learn (and to make mistakes). For Problem 2, the worked solution consisted of 7 steps:

1. Categorize the problem as a distance-velocity problem under constant acceleration
2. Instantiate the distance-velocity equation ($V_f^2 = V_i^2 - 2ad$)
3. Given the projectile motion and lack of atmosphere, infer that the acceleration of the ball is equal to the

[1] These levels are from a 10-level catalog of transfer tasks used in DARPA's Transfer Learning Program (http://fs1.fbo.gov/EPSData/ODA/Synopses/4965/BAA05-29/BAA05-29TransferLearningPIP.doc)

```
…
(groundOf Planet-1 Ground-1)
(performedBy Throwing-1 Astronaut-1)
(no-GenQuantRelnFrom
          in-ImmersedFully Planet-1 Atmosphere)
(eventOccursNear Throwing-1 Ground-1)
(objectThrown Throwing-1 Ball-1)
(querySentenceOfQuery Query-1
   (valueOf (AccGravityFn Planet-1) Acc-1))
…
```

Figure 2: Part of Problem 2 representation

acceleration due to gravity (a = g)

4. Because of the projectile motion event, the ball is not moving at the maximum height ($V_f = 0$ m/s)
5. Solve the equations for the acceleration due to gravity ($g = -1.6$ m/s^2)
6. Sanity check the answer (the answer is consistent)
7. Select the appropriate multiple choice answer ("c")

Figure 3 shows how Step 3 is represented. We store the worked solution along with the problem description as a case in our case library, which will be used to solve new problems. This is a very simple form of learning, learning by accumulating examples. While simple, it is very powerful, as our experiment below illustrates. We discuss our plans to move beyond this later.

Solving a Problem

Problems are presented as cases of predicate calculus facts. The first phase of problem solving is to generate an analogy with a relevant example. This is done in three steps. First, our system retrieves a worked solution from the case library using MAC/FAC, producing a mapping. If the mapping between the worked solution and the problem does not include all the event structure, i.e. facts relating to the events of the problem, then the system will create a new probe consisting only of the unmapped events in the problem case. This process continues until there is no more unmapped event structure or the retrieval fails to find matches for the remaining unmapped event structure.

```
(stepType Step3 DeterminingValueFromContext)
(stepUses Step3 (isa Throwing-1 ThrowingAnObject))
(stepUses Step3 (occursNear Throwing-1 Ground-1))
(stepUses Step3
  (no-GenQuantRelnFrom
     in-ImmersedFully Planet-1 Atmosphere))
(stepUses Step3 (objectMoving Upward-1 Ball-1))
…
(stepUses Step3 (direction Upward-1 Up-Directly))
(solutionStepResult Step3
  (valueOf
   (AtFn ((QPQuantityFn Speed) Ball-1)
         (EndFn Upward- 1))
   (MetersPerSecond 0)))
```

Figure 3: Problem 2 Worked Solution Step 3

(This is important for handling composing problems.) After the retrieval stage is complete, the system proceeds with problem solving using the candidate inferences produced by the analogs as necessary, which includes worked solution steps.

There are several different broad types of problems in the AP Physics exam, including deriving the value of a quantity and determining what situation would be consistent with a given set of numerical values. The core of each of these problem types is determining the value of quantities. The system begins by categorizing the problem and determining which quantity or quantities should be solved for. This is done through rules which analyze the fact indicating the query of the problem. The system solves for quantities in three ways. First, it may already be known as part of the problem. Second, the candidate inferences of the mapping may contain an applicable solution step in which the quantity was assumed in the analog. Third, the candidate inferences might indicate a relevant equation containing the sought quantity. In this case, the system first looks for values for the other quantities in the equations, and then attempts to solve the equation for the original parameter. The algebra routines are straightforward, based on the system in Forbus & de Kleer (1993) and currently not extendable by learning.

To determine whether or not a solution step suggested by candidate inferences is valid, its context is checked in the worked solution. Suppose the step assumes that the acceleration of a rock in freefall is 10 m/s^2, because the rock is falling on Earth and there is no air resistance. To apply this step, the system must be able to infer that there is no air resistance in the current situation and that the event occurs on Earth. This verification step helps guard against inappropriate applications of candidate inferences.

It is important to note that the system does not start with any of the equations of physics – it only has access to them through examples of how they have been used in the worked solutions. Thus, analogical reasoning is essential to a Companion's ability to solve any problems.

Before selecting a multiple choice answer, the system looks for any candidate inferences indicating a sanity check was used in the mapped worked solution. For example, if a problem asked, "How far a ball would fall off a 200m building in 4s?" there would be a sanity checking step in which the computed answer, 80m, was compared to 200m. When this worked solution is used in solving future problems, the analogy produces candidate inferences indicating the type of check and corresponding quantities in the current problem that are involved. Currently, we only employ this check if the quantity sought for is involved in the comparison. This is because it is clear how to resolve a failure, i.e. use the value compared against it instead, because it constitutes a limit point (Forbus 1984) for that quantity[2]. In other cases what to do is much less clear, and we plan to learn rules for resolving such

[2] This heuristic is reasonable for mechanics but would not be appropriate for other domains, such as thermodynamics.

problems in the future.

After an answer is found to be consistent, it is compared against each of the answer choices. The system selects either the closest answer for quantity value questions or the consistent answer choice in a qualitative behavior problem, such as Problem 3 in Figure 1.

Model formulation via analogy

Solving physics problems typically requires four kinds of modeling decisions and assumptions. (1) Which equations are applicable for a given situation. Even in a relatively constrained domain like physics, the number of potentially relevant equations can be quite large due to specialized forms. (2) Some parameter values are inferred from circumstances. For instance, Problem 2 is not solvable if the system fails to recognize that the ball's velocity at the top of the projectile motion event is zero. (3) Some circumstances are assumed by default. The most common of these in AP Physics is to assume that events happen on Earth and are subject to Earth's gravity unless otherwise mentioned. (4) Simplifying assumptions, such as viewing an object as a point mass or assuming a collision is elastic, are often required for tractability.

Three of the four types of modeling assumptions are handled by our system directly through analogical reasoning. That is, relevant equations, determining parameter values, and default circumstances are handled directly by the analogy with the worked solution. Only the fourth type, categorizing an everyday object in terms of an abstraction, is not currently handled by our system. Instead, we take the categorization as acceptable if it is compatible with the rest of the mapping. This works well when the analogous problems are close, but we expect to run into trouble when the analogs are more distant.

Learning conditions for such categorizations is one of our goals, but it turns out to be complex. Worked solutions provide little information about why a modeling assumption they used is reasonable. For example, modeling the ball as a point mass in Problem 2 is not even mentioned in the worked solution. Students must generalize from a body of examples they have seen to learn when to apply such ideas. We conjecture that this is because the ontology of everyday things is very broad, and the subsets of object types that are appropriate for a particular idealization are not tightly localized to one part of the ontology. For example, rocks, coins, soda cans, and ferrets can all be considered as point masses for some kinds of problems, but most ontologies would not consider these categories as being particularly close otherwise. Furthermore, the idealizations are very context dependent. A coin falling off a building could be considered a point mass, but to model the exact same coin spinning on a table as a point mass would be a mistake. On the other hand, applying modeling knowledge via within-domain analogies turns out to be quite robust, as our experiment illustrates.

An Experiment

An experiment was conducted to evaluate a Companion's ability to transfer knowledge. The evaluation was carried out by the Educational Testing Service, who remotely accessed a Companion running on our cluster.

Methodology

Each training set consisted of 5 quizzes, each containing a variation of the four problem types illustrated in Figure 1, for a total of 20 problems. After a quiz was administered, the Companion would get the worked solutions for the problems on the quiz. Thus the worked solutions from earlier quizzes were available for use in solving later quizzes within the training set. To ascertain whether the Companion could transfer what it learned, for each original training set, 30 additional transfer training sets were created, five for each transfer level. The structure of the transfer training sets was the same as the original training set minus one quiz, e.g., four quizzes of four problems each. Thus the Companion would be run with the training set followed by the transfer training set (the *transfer* condition), after which its memory would be reset. Then it would be run again with just the transfer training set (the *no-transfer* condition). Comparing the learning curves for these two conditions provides a measure of how much was learned via transfer.

There are three ways for transfer to manifest itself. (1) The system may get a jump start, i.e., the learning curve in the transfer condition has a higher y-intercept. (2) The system may learn faster. (3) The system may reach a higher level of performance. These are not mutually exclusive. Given the rapidity of analogical learning, we were most interested in jump start.

Results

Figure 4 shows the learning curves for both the transfer and no-transfer conditions for each transfer level. All levels showed a statistically significant jump start ($p<.01$). For TL-1, TL-4, and TL-5, the jump start was 88 percent. Other levels were not as high: TL-2 was 50 percent, TL-3 was 25 percent, and TL-6 was 44 percent.

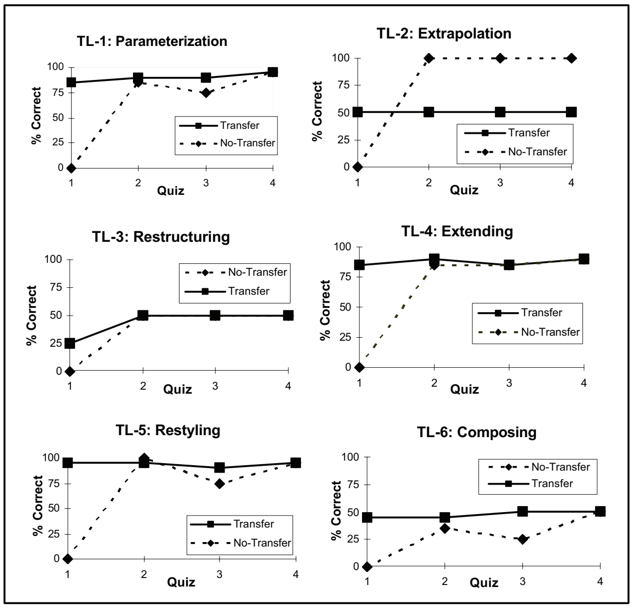

Figure 4: Experiment Results

While the jump start results support our hypothesis, there are some results that require further explanation. First, in TL-2, there was apparent negative transfer; the no-transfer condition out-performed the transfer condition. This occurred because the Companion was repeatedly getting a multiple choice question correct for the wrong reasons. An error in the worked solution representations for problem type 3 caused the Companion to incorrectly assume a value for acceleration, which coincidently led to the correct answer, in the no-transfer condition. This could not be corrected given the external nature of this evaluation.

Second, there are low ceilings in TL-3 and TL-6, where the Companion was unable to score above 50 percent on any of the quizzes. A careful analysis of these problems indicates that they are due to limitations in the Companion's strategies, rather than any issues with the analogical reasoning portions of the system. For Problem Type 3, the Companion does not handle "plugging in" different parameter values for each answer choice efficiently enough to prevent timeouts. The low scores on TL-6 are because the Companion's strategies assume that a given problem either demands numerical values or symbolic values, but not both, and thus it could not handle a composition of a symbolic problem with a numerical problem. Given our current focus on learning domain knowledge rather than strategies, the current system's behavior cannot improve to overcome such problems. This is one reason why we are expanding our investigations to include strategy learning in future work.

Related Work

There have been several explorations of solving textbook problems. Within qualitative reasoning, de Kleer's (1977) pioneering work in reasoning about sliding motion problems demonstrated that qualitative reasoning was required for solving many quantitative mechanics problems. The majority of the work on physics problem solving has focused on equation search and solving. Two such systems, Mecho (Bundy 1979) and Isaac (Novak 1977), take natural language input and move to structural abstractions via rule-based systems to solve the problems. In contrast, our work uses analogy to apply modeling assumptions and relevant equations. The HALO project (Barker *et al.* 2004) built knowledge-based systems that solved AP Chemistry problems based on 5 pages of textbook knowledge. Like the HALO project, our system was evaluated on unseen problems administered by non-developers. None of these efforts address learning, whereas learning domain knowledge is our central focus.

The relevant AI research on analogy in problem solving includes (Melis & Whittle 1999; Veloso & Carbonell 1993). The closest systems to ours are Cascade (VanLehn 1998) and APSS (Ouyang & Forbus 2006). Both Cascade and APSS start with equations and other domain knowledge, and use analogy primarily for search control. Our use of analogy differs from Cascade and APSS by learning domain knowledge.

There has been an increasing interest in transfer learning recently. Lui and Stone (2006) use a version of SME to accelerate learning of state action policies in novel, but similar, tasks within keep-away soccer. Instead of using structure mapping to accelerate learning, we use structure mapping as our learning mechanism. Sharma *et al.* (2007) use reinforcement learning for credit assignment and case-based reasoning to learn value functions for variations in a real-time strategy game. As in our work, similarities between the current situation and previous cases drive knowledge transfer.

Discussion

This paper has examined how a Companion can use analogy to go from the unruly, broad common sense world to the refined world of parameters, equations, and modeling assumptions. While the overall performance is already quite good, it should be noted that only represents roughly 20% of the material in the Newtonian Mechanics portion of the AP Physics exam. Our future work is motivated by the goal of expanding the system to the point where it can learn all of the material on an AP Physics exam, which is even broader than Mechanics.

In addition to testing the system on more problem types, there are certain additions that we believe to be essential to handle more complex transfer. First, we plan to move beyond learning by accumulating examples. We plan to construct generalizations using SEQL (Kuehne *et al.* 2000) to facilitate the Companion's ability to transfer what it learns more broadly. Also, equations might be learned as *encapsulated histories* (Forbus 1984), which, being parameterized, could extend a Companion's reach still further. Second, as Companions accumulates generalizations in one area of physics, we will explore how dynamical analogies (Olsen 1966) can facilitate transfer learning into other areas of physics, such as electrical and hydraulic domains. Cross domain analogies, while risky, accelerate one's understanding of a new domain. Because of this, we plan on increasing interactivity so that advice such as "heat flow is like water flow" can be understood and leveraged by a Companion.

Acknowledgments

This research was supported by DARPA under the Transfer Learning program. We thank Catherine Trapani and Vincent Weng at the Education Testing Service for administering the evaluation and Cynthia Matuszek, Blake Shepard, and Casey McGinnis at Cycorp for providing the testing materials.

References

Barker, K., Chaudhri, V., Chaw, Yi, S., Clark, P., Fan, J., Israel, D., Mishra, S., Porter, B., Romero, P., Tecuci, D., and Yeh, P. 2004. A Question Answering System for AP Chemistry: Assessing KR Technologies. *Proc. of the 9th International Conference on Knowledge Representation and Reasoning*.

Bundy, A. 1979. Solving mechanics problems using meta-level inference. *Proc. of IJCAI-79*.

de Kleer, J. 1977. Multiple representations of knowledge in a mechanics problem solver, pp.299–304. *Proc. of IJCAI-77*.

Falkenhainer, B. 1992. Modeling without amnesia: Making experience-sanctioned approximations. *Proc. of QR02.*

Falkenhainer, B., Forbus, K. and Gentner, D. 1989. The Structure-Mapping Engine: Algorithm and examples. *Artficial Intelligence.* 41.

Falkenhainer, B. & Forbus, K. 1991. Compositional modeling: finding the right model for the job. *Artificial Intelligence* 51.

Flores, J. & Cerda, J. 2000. Efficient modeling of linear circuits to perform qualitative reasoning tasks. *AI Communiations*, 13(2).

Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Forbus, K. 2001. Exploring analogy in the large. In Gentner, D., Holyoak, K., and Kokinov, B. (Eds.) *Analogy: Perspectives from Cognitive Science.* MIT Press.

Forbus, K. & de Kleer, J. 1993. *Building Problem Solvers*, MIT Press.

Forbus, K., Ferguson, R., and Gentner, D. 1994a. Incremental structure-mapping. *Proc. of CogSci-1994.*

Forbus, K., Gentner, D., and Law, K. 1994b. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19.

Forbus, K. & Hinrichs, T. 2006. Companion Cognitive Systems: A step towards Human-Level AI. *AI Magazine*, 27(2)

Forbus, K. & Oblinger, D. 1991. Making SME greedy and pragmatic. *Proc. of CogSci-1990*

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

Klenk, M., Forbus, K., Tomai, E., Kim,H., and Kyckelhahn, B. 2005. Solving Everyday Physical Reasoning Problems by Analogy using Sketches, *Proc. of AAAI-05.* Pittsburgh, PA.

Kuehne, S., Forbus, K., Gentner, D., and Quinn, B. 2000. SEQL: Category learning as incremental abstraction using structure mapping. *Proc. of CogSci-2000.*

Liu, Y., & Stone, P. Value-Function-Based Transfer for Reinforcement Learning Using Structure Mapping. *Proc. of AAAI-06.* Boston, MA.

Melis, E. & Whittle, J. 1999. Analogy in inductive theorem proving. *Journal of Automated Reasoning*, 22. Kluwer.

Nayak, P. 1994. Causal approximations. *Artificial Intelligence.* 70.

Novak, G.S. 1977. Representation of Knowledge in a Program for Solving Physics Problems. *Proc. of IJCAI-77.*

Olson, H. 1966. *Solutions of Engineering Problems by Dynamical Analogies*, D. Van Nostrand.

Ouyang, T. & Forbus, K. 2006. Strategy variations in analogical problem solving. *Proc. of AAAI-06.*

Rickel, J. & Porter, B. 1994. Automated modeling for answering prediction questions: selecting the time scale and system boundary. *Proc. of AAAI-94*.

Sharma, M., Holmes, M., Santamaria, JC., Irani, A., Isbell, C., and Ram, A. 2007. Transfer Learning in Real-Time Strategy Games Using Hybrid CBR/RL *Proc. of IJCAI-07.*

VanLehn, K. 1998. Analogy Events: How Examples are Used During Problem Solving. *Cognitive Science* 22(19): 347-388.

Veloso, M. & Carbonell, J. 1993. Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning*, 10.