Title
Internal Analogy: A Model of Transfer within Problems

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 12(0)

Authors
Hickman, Angela Kennedy
Larkin, Jill H.

Publication Date
1990

Peer reviewed

Internal Analogy: A Model of Transfer within Problems

Angela Kennedy Hickman
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
ach@ml.ri.cmu.edu

Jill H. Larkin
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213
larkin@tulip.psy.cmu.edu

Abstract

Understanding problem solving and methods for learning is a main goal of cognitive science. Analogical reasoning simplifies problem solving by transferring previously learned knowledge from a source problem to the current target problem in order to reduce search. To provide a more detailed analysis of the mechanisms of transfer, we describe a process called *internal analogy* that transfers experience from a completed subgoal in the same problem to solve the current target subgoal. We explain what constitutes an appropriate source problem and what knowledge to transfer from that source, in addition to examining the associated memory organization. Unlike case-based reasoning methods, this process does not require large amounts of accumulated experience before it is effective; it provides useful search control at the outset of problem solving. Data from a study of subjects solving DC-circuit problems designed to facilitate transfer supports the psychological validity of the mechanism.

1. Introduction

Analogical reasoning is an effective method of recycling past experience to guide problem solving. To begin the analysis of this process, we formulate the following five steps. First, the problem solver must determine a set of candidate sources. The utility of the procedure relies on the identification of the relevant knowledge. Second, one solution must be retrieved to function as the actual source. Third, the source solution must be reinstantiated and modified to solve the target problem. Fourth, the new solution should be stored so that the problem solver can reason from it to solve future problems. Fifth, the problem solver should receive some knowledge of results concerning the effort required to perform the analogy and the successfulness of the procedure. This information can be used to provide feedback to the retrieval steps of the process.

Although analogy has been explored previously [1, 2, 7, 8, 16, 17], most of the work has focused on the mapping procedure outlined in step three above. [11], [12] and [19] have addressed memory organization (related to steps one, two, and four above) in detail, but their ideas have not been integrated with an analogical mechanism. ICARUS [14] and EUREKA [10] have incorporated the whole process to some extent. However, neither system embeds its solution within a general implementation of a problem solver (for example, EUREKA is not capable of backtracking). As a result, they cannot solve problems which are as difficult as those reported in this study.

This paper addresses the first four steps of the process with a transfer mechanism called *internal* analogy, which works on similar subgoals of a single problem. This is in contrast to *within-domain* and *cross-domain* analogy which transfer knowledge across separate problems from the same domain and different domains, respectively. Work in progress using derivational analogy [4] in PRODIGY [3, 22] is beginning to address most of the steps above, but for cross-problem, within-domain analogy.

Our internal analogy mechanism is tightly integrated into a general problem solver for the physical sciences, RFERMI, and is effective in reducing search [9]. Unlike case-based reasoning methods, the process does not require large amounts of accumulated experience before it is effective; it provides useful search control at the outset of problem solving. In addition, psychological predictions drawn from the computational model of internal analogy were supported by data from a study of subjects solving DC-circuit and fluid statics problems that were designed to facilitate transfer.

The next section presents the implementation of the internal analogy algorithm in RFERMI, as well as an example trace of the non-learning system. Section 3 contains the psychological predictions derived from the computational model and the analysis of the data we collected. We conclude with a discussion evaluating internal analogy.

2. Computational Model

The internal analogy process described in the preceding section is implemented in a problem solver for the physical sciences named RFERMI. This system is a rule-based version of the FERMI system [15] and is based in part on studies of effective representations and methods for solving physics problems [5, 18]. Its task domains range from linear algebra and DC-circuits to fluid statics and classical mechanics. RFERMI maintains a principled decomposition of knowledge in order to retain the power of its domain specific knowledge while utilizing the cross-domain applicability of its more general knowledge.

RFERMI's declarative knowledge of scientific principles is organized in a *quantity hierarchy* which is stored as frames. The frame system used to implement the hierarchy is a component of FRulekit [20], a forward chaining production system. Through the use of inheritance, this hierarchy efficiently stores knowledge about quantities such as resistances, pressure drops, potential drops, and two dimensional areas.

RFERMI's procedural knowledge, as organized in its *actions hierarchy*, is of two types. First, domain specific knowledge is stored in *puller* frames that are interpreted into FRulekit rules. These pullers encode equational knowledge, such as Ohm's law, and procedures, such as those for finding the pressure drop between two points in a static liquid or the electro-magnetic force of a battery. Second, RFERMI's more general and widely applicable knowledge, such as its iterative decomposition procedure, constitute its *methods*. Methods are associated with generalized quantities so that a quantity inherits access to a method from superordinate quantities. For example, potential drop inherits access to the path invariance method from scalar field difference, which is a generalized quantity. Therefore, the method for equating potential drops along two alternate paths can be used to solve for a potential drop.

Although the system is implemented in a forward chaining production system, it maintains a backward chaining control structure via the goal monitor. The space it searches while solving for an unknown quantity is structured in a traditional AND-OR manner. When there are multiple means for pursuing a goal (i.e., pullers and methods) an OR node is generated, and when a method spawns a conjunctive set of unmet subgoals (i.e., unknown quantities) an AND node is generated. RFERMI organizes its search in a manner combining depth-first and breadth-first expansion. At OR nodes, it contains heuristic preferences for using the specific puller knowledge over the more general methods.

As an example, we present RFERMI's problem solving without learning on the problem in Figure 2-1(see Figure 2-2). The system first solves for I_2 by applying a puller. Lines 7-14 show that it solves for R_1 by generating subgoals for V_{ab} and I_1. V_{ab} is solved using a puller, which results in R_1's solution. In line 16, RFERMI chooses a different puller to solve I_3 than it used to solve I_1. (It could reduce the required search by solving I_3 in a similar manner since the goals are analogical and the system solved I_1 with relatively little effort.) In both cases, both pullers were applicable, and the system chose randomly. However, this line of reasoning ends in goal circularity (see line 20) since V_{cd} has already been posted. In lines 21-23, it tries another instantiation of the same puller, which also results in goal circularity. Finally, in lines 24-29, it obtains the value of V_{cd} by using the puller that solved I_1. The end of the trace shows how R_4 is solved.

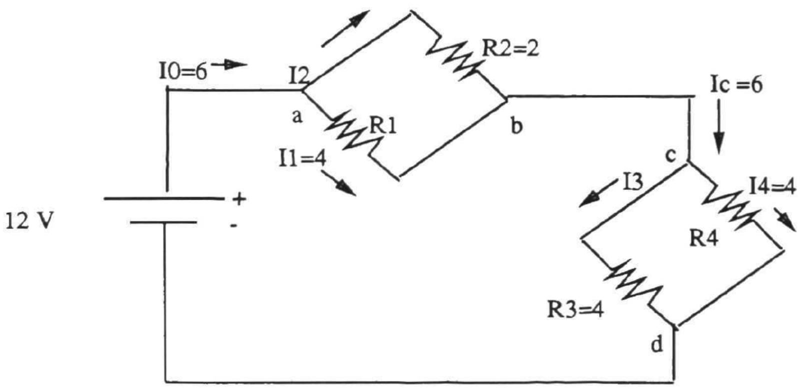

Figure 2-1: Problem 5.

The implementation of internal analogy in RFERMI relies on the trace facility of the problem solver. While solving a problem, the system stores the expanded goal tree along with the status of each goal (succeeded, failed, or pending). For goals that succeed, the instantiated productions that achieved the goal are also stored. As shown by the internal analogy algorithm in Figure 2-3, when the system begins to solve a new target subproblem, it checks if it has previously solved a goal of the same quantity type. (For example, I_1 is of the same quantity type as I_3 in Figure 2-1 since they are both currents.) If so, the most recently solved

1. Post Goals:	$\mathbf{I_2, R_1, I_3, R_4}$	
2. Choose Goal:	$\mathbf{I_2}$	
3. Apply Puller:	$I_2 = I_0 - I_1$	current flowing into a = current flowing out of a
4. Lookup:	$I_0 = 6$	
5. Lookup:	$I_1 = 4$	
6. Compute:	$I_2 = 2$	
7. Choose Goal:	$\mathbf{R_1}$	
8. Apply Puller:	$R_1 = V_{ab} / I_1$	Ohm's law (V=IR)
9. Apply Puller:	$V_{ab} = I_2R_2$	Ohm's law (V = IR)
10. Lookup:	$I_2 = 2$	
11. Lookup:	$R_2 = 2$	
12: Compute:	$V_{ab} = 4$	
13. Lookup:	$I_1 = 4$	
14. Compute:	$R_1 = 1$	
15. Choose Goal:	$\mathbf{I_3}$— internal analogy would apply	
16. Apply Puller:	$I_3 = V_{cd} / R_3$	Ohm's law (V = IR)
17. Apply Puller:	$V_{cd} = I_4R_4$	Ohm's law (V = IR)
18. Lookup:	$I_4 = 4$	
19. Apply Puller:	$R_4 = V_{cd} / I_4$	Ohm's law (V = IR)
20. Fail Circular Goal:	V_{cd}	
21. Apply Puller:	$V_{cd} = I_3R_3$	Ohm's law (V = IR)
22. Apply Puller:	$I_3 = V_{cd} / R_3$	Ohm's law (V = IR)
23. Fail Circular Goal:	V_{cd}	
24. Apply Puller:	$I_3 = I_c - I_4$	current flowing into c = current flowing out of c
25. Lookup:	$I_c = 6$	
26. Lookup:	$I_4 = 4$	
27. Compute:	$I_3 = 2$	
28. Lookup:	$R_3 = 4$	
29. Compute:	$V_{cd} = 8$	
30. Lookup:	$R_3 = 4$	
31. Compute:	$I_3 = 2$	
32. Choose Goal:	$\mathbf{R_4}$— internal analogy would apply	
33. Apply Puller:	$R_4 = V_{cd} / I_4$	Ohm's law (V = IR)
34. Lookup:	$V_{cd} = 8$	
35. Lookup:	$I_4 = 4$	
36. Compute:	$R_4 = 2$	

Figure 2-2: RFERMI's behavior on Problem 5 with no learning.

instance is chosen as the *candidate* source. If the candidate source succeeded and no more information was known about it than is known about the current problem, then it is chosen as the *actual* source. Otherwise, it is rejected because the system had additional knowledge during the previous problem solving which may have been crucial to its success. Without that knowledge, it may be unable to recycle the old solution to solve the current target problem. The failure case is decided in just the opposite manner. If the candidate source failed and contained no less information than the current problem, it is chosen. Otherwise, it is rejected. This check ensures that RFERMI does not choose a source that failed because there was less information available to the problem solver and prematurely fail the current target as a result. The mapping proceeds based on the success or failure of the source. If the source failed, then the processing of the current goal is suspended and another goal is explored. If all other problem solving fails, this goal may be later reopened. If, on the other hand, the source succeeded, its solution is appropriately reinstantiated for the current subgoal, and the solution is replayed. Since this newly solved subgoal is contained in the current problem, it is available to the algorithm as a future candidate source.

```
1. IF there are untested previously explored subgoals of the same type as target
2.         THEN candidate := most recently explored same-type subgoal
3.         ELSE fail internal-analogy
4. IF candidate succeeded
5.         THEN IF information-content(candidate) <= information-content(target)
6.                     THEN source := candidate & reinstantiate the source solution
                       to solve the target
7.                     ELSE tested(source) := TRUE & internal-analogy(target)
8.         ELSE IF information-content(candidate) >= information-content(target)
9.                     THEN source := candidate & suspend(target)
10.                    ELSE tested(source) := TRUE & internal-analogy(target)
```

Figure 2-3: The internal analogy algorithm.

Steps 5 and 8 of the algorithm in Figure 2-3 compare the amount of information known about a candidate goal at the time it was solved with the amount of information known about the current target goal. This comparison is carried out in RFERMI by calculating the set of variables in the left hand sides of the rules that solved the candidate source goal and that had known values. We call this the *information content* of the candidate source. The information content of the current target goal is computed by calculating the set of these same variables that have known values in the current working memory. For example, suppose the candidate source goal is to find the potential drop between points *a* and *b* in a circuit, and the resistance and the current between those two points were known. Now suppose the current target goal is to find the potential drop between points *c* and *d* in the same circuit, and the equation that solved the candidate source goal was potential-drop = current * resistance. If the resistance between *c* and *d* is known but the current is not, then the information content of the candidate source goal is said to be greater than that of the target goal (because the current was known in the candidate source goal).

Step 6 of Figure 2-3 reinstantiates and replays the solution of the source problem in order to solve the target problem. RFERMI carries out this step by instantiating the productions that solved the source goal in the current working memory and applying them. We call this *operator-driven mapping*, and it answers the important question that Structure Mapping [7] inadvertently poses: how does one identify the salient structure to map? We operationally define the salient structure to be the relevant variables that are tested in the left-hand sides of the operators that solved the source goal.

As a last comment, we mention that it is especially important to only suspend the processing in Step 9 of Figure 2-3. The system cannot terminally fail the goal because one of RFERMI's other general methods may still solve the problem, although more expensively.

The internal analogy mechanism described above has proven to be an effective learning mechanism in RFERMI. Detailed theoretical and empirical analyses of the search reduction it provides are described in [9].

3. Protocol Data

If, as hypothesized, the internal analogy mechanism embedded in RFERMI has any psychologically validity, then the computational model described in the previous section predicts that subjects will exhibit the following behaviors during problem solving:

- Knowledge will be transferred from either previously successful or previously failed goals.
- The source will be of the same quantity type as the target and have a compatible information content.
- Problem solving using analogy will require less effort (search) than would otherwise be necessary.
- Transfer from previously failed problem solving will enable the subject to know that a particular procedure as instantiated at the current point will fail. Thus, he should choose a different procedure.

- Transfer from previously successful problem solving will allow the subject to know precisely which procedures to choose to calculate the quantity and all its subquantities. For problems in the physical sciences, this means that the subject should know which equations to reuse to solve the unknown. However, these equations must be reinstantiated to reflect the new problem solving context.
- Since RFERMI randomly selects among its applicable pullers for solving any given subgoal, we predict that subjects will show individual differences in their problem solving behavior. The system also has two different control strategies: one strictly depth first and the other more breadth first. It's problem solving differs according to the current control strategy. As a result, we predict further individual differences will arise from the subjects' varied control strategies.

To test the predictions, we studied subjects solving problems from two of RFERMI's task domains, DC-circuits and fluid statics. These problems were designed to facilitate three kinds of transfer: internal analogy, within-domain analogy, and cross-domain analogy. The four subjects had all earned an A or a B in a year-long college physics course, but they had not solved any problems in these domains for several years. We chose subjects with this level of proficiency because we believed that they would be the most likely to exhibit the desired transfer. Subjects with a high level of expertise tend to use compiled knowledge rather than analogical reasoning; subjects with very little expertise tend to use brute force search. The subjects were given a remedial, which was in a two-column format, covering the knowledge necessary for the experiment. The left column contained the circuit information, and the right column contained the fluid statics information. Analogical concepts were presented directly across the page from each other. In order to verify the remediation, the subjects were asked to explain sparse written solutions to three example problems. These example problems were also designed to serve as analogical sources for the five problems that the subjects were asked to solve next.

We observed all three types of transfer. However, the more local types of transfer happened more frequently; only one instance of cross-domain transfer occurred. Due to space limitations, we discuss the subjects' behavior only on Problem 5, which was shown in Figure 2-1 and designed to facilitate internal analogy. Below we demonstrate that RFERMI with the internal analogy mechanism models the subjects' behavior well. The mechanism also reduces the search previously required to solve I_3 and R_4 by about 50% (compare lines 15-36 of Figure 2-2 with lines 16-30 of Figure 3-1).

We begin by comparing the behavior produced by RFERMI with internal analogy to that of Subject 1 on the example problem (see Figure 3-1). Problem solving for both proceeds similarly, except for the following differences which are unimportant with respect to the analogical mechanism. Between lines 2 and 3 of the figure, Subject 1 does some erroneous problem solving and decides to start over. As can be seen in lines 3 and 18 of the protocol, the system always posts an equation with the desired unknown on the left hand side, while the subject posts the version of the current invariance equation that corresponds to the associated prose in the remedial. In lines 8 and 10 of the protocol, Subject 1 posts incorrect equations; this will have interesting side effects later in the problem solving. The problem solver represents potential drops as a drop between two points, regardless of the path. The subject, however, clearly distinguishes potential drops with the same endpoints over different paths; this leads to his extra step in line 9. Occasionally, Subject 1 will take an arithmetical "shortcut" by not restating the implied left-hand side of the equation or by reducing fractions to their lowest terms (lines 11-12, 26-27 and 29).

Ignoring these small differences, RFERMI models the subject extremely well. Both solve equations for I_2 and R_1 in a straightforward manner. At line 17, the system's analogical mechanism is invoked because it is has solved a goal of the same quantity type with a compatible information content, I_2 (I_0 and I_1 are known while I_2 is unknown at line 3, and I_c and I_4 are known while I_3 is unknown at line 18). It retrieves the productions that solved I_2 and reinstantiates them. This saves the system search time in two ways. First, it does not have to compute which productions to apply— the analogy mechanism specifies them. Second, there are other applicable pullers at this point that would require more problem solving effort if they were used, as shown in lines 15-31 of Figure 2-2. Subject 1 also recognizes that I_3 is an analogical goal to I_2 at line 17: he states, "this (pointing to I_3) is just like that (pointing to I_2)". Then, he quickly reinstantiates the equation that he used in line 3 and solves for I_3. Similar recycling of past experience occurs for both the problem solver and the subject in lines 22-30 while solving for R_4. When Subject 1 says, "back to here," in these lines, he is pointing to the equation $R_1 = I_1/V_{r1}$ which he wrote in line 8. The subject analogizes from the incorrect equations in lines 8 and 10 and reuses them in lines 24 and 25, which causes him to derive an incorrect answer for R_4. Had he used the correct equations earlier, he would have solved this problem

	FERMI	Subject 1	
1. Post Goals:	**I_2, R_1, I_3, R_4**	**I_2, R_1, I_3, R_4**	
2. Choose Goal:	**I_2**	**I_2**	
3. Apply Puller:	$I_2 = I_0 - I_1$	$I_0 = I_1 + I_2$	
4. Lookup:	$I_0 = 6$	$I_0 = 6$	
5. Lookup:	$I_1 = 4$	$I_1 = 4$	
6. Compute:	$I_2 = 2$	$I_2 = 2$	
7. Choose Goal:	**R_1**	**R_1**	
8. Apply Puller:	$R_1 = V_{ab} / I_1$	$R_1 = I_1 / V_{r1}$	
9. Apply Method:		$V_{r1} = V_{r2}$ [1]	
10. Apply Puller:	$V_{ab} = I_2R_2$	$V_{r2} = I_2 / R_2$	
11. Lookup:	$I_2 = 2$	$= 2/$	
12. Lookup:	$R_2 = 2$	2	
13. Compute:	$V_{ab} = 4$	$V_{r1} = 1$	
14. Lookup:	$I_1 = 4$	$I_1 = 4$	
15. Compute:	$R_1 = 1$	$R_1 = 4$	
16. Choose Goal:	**I_3**	**I_3**	
17. Analogize:	*Fires analogy to I_2*	*"This is just like that."*	
18. Apply Puller:	$I_3 = I_c - I_4$	$I_c = I_3 + I_4$	
19. Lookup:	$I_0 = 6$	$I_c = 6$	
20. Lookup:	$I_4 = 4$	$I_4 = 4$	
21. Compute:	$I_3 = 2$	$I_3 = 2$	
22. Choose Goal:	**R_4**	**R_4**	
23. Analogize:	*Fires analogy to R_1*	*"Back to here."*	
24. Apply Puller:	$R_4 = V_{cd} / I_4$	$R_r = I_4 / V_{r4}$	
25. Apply Puller:	$V_{cd} = I_3R_3$	$V_{r3} = I_3/R_3$	
26. Lookup:	$I_3 = 2$	$= 1/$	
27. Lookup:	$R_3 = 4$	2	
28. Compute:	$V_{cd} = 8$	$V_{r4} = 1/2$	:Apply Method
29. Lookup:	$I_4 = 4$	$4/.5 = 8$	:Compute
30. Compute:	$R_4 = 2$	$R_4 = 8$	

Figure 3-1: The behavior of RFERMI with learning and Subject 1 on Problem 5.

correctly.

All four subjects performed internal analogy on Problem 5, but each exhibited a different control structure. Subject 1 backward chained much like RFERMI, while Subject 4 demonstrated more expertise in the domain and forward chained. This behavior is consistent with the results reported in [21] that show that experts tend to forward chain in search spaces that they expect to be small. Subject 2 began by backward chaining and switched to forward chaining as he gathered more experience in the domain. Subject 3 struggled to complete the problem and explored the subgoals in a nonstandard order. We now examine each of the other subject's problem solving more closely.

In contrast to Subject 1, Subject 4 solves the problem by forward chaining. This subject maps the

[1]The potential drop between any two points is the same regardless of the path chosen between the points.

analogical subgoals explicitly by their quantity type and information content. At the outset of the problem solving, he says, "So, I have two resistors where the current is given and the resistance is left unknown (R_1 and R_4) and two resistors where the resistance is given and the current is left unknown (R_2 and R_3)". He proceeds to solve for I_2 and R_1. At this point he says, "similar situation here," and solves for I_3, reusing the the equation that solved I_2 reinstantiated for the current goal. In a similar fashion, he uses the equations that solved R_1 to solve R_4. This subject, like Subject 2 and Subject 1 on other problems, tends not to verbalize the uninstantiated equation during the replay of the analogy but verbalizes the instantiated form instead.

Subject 3, who begins backward chaining and switches to forward chaining, states early in his problem solving that $I_1=I_4=4$ and $I_0=I_c$. When he solves I_2, he immediately states the same answer for I_3, without additional computation. It appears that his analogical reasoning is more advanced than RFERMI's. In addition to reposting and reinstantiating equations, this subject is able to recognize when the relevant variables have exactly the same value, and the answer can be recycled directly. With this straightforward extension added to the system, it could gain an even greater reduction in search. The point at which Subject 3 switches to forward chaining is also significant: he finishes solving for V_{ab}, and he recycles the equation he used in a newly instantiated form for V_{cd}. The switch from backward chaining to forward chaining seems to be triggered, at least in part, by an analogical goal.

Subject 2 is considerably less skilled at solving these types of problems than the other subjects. He struggles to solve any of the subgoals using the same knowledge encoded in RFERMI's pullers, methods, and algebra module. When he does finally solve I_2 and R_1, however, he immediately restates the current invariance relation and quickly solves I_3 and R_4. There is nothing in his analogical transfer that we did not observe in the previous three protocols.

4. Discussion and Conclusions

Although the subjects showed individual differences in their control strategies, the basic components of the analogical reasoning were those that the computational mechanism predicted. The subjects transferred knowledge from successful problem solving in order to reduce the effort required to solve the target subgoal. They simply reposted the previously successful equations and reinstantiated them in the current context. In every instance, the sources and targets were of the same quantity type and had compatible information contents.

The system models Subject 1 well in its current state. With a forward chaining control strategy, it could easily model Subject 4 as well. To model Subject 2, one additional capability must be added to the system: it should recognize those occasions when the equations need not be reinstantiated but the value may be directly recycled. Even though the computational model focused our attention toward particular problem solving behavior in the protocols, the protocols continue to suggest useful extensions to the system.

Relaxing the notion of compatible information content will provide internal analogy with a more flexible matching mechanism than either SOAR's chunking [13] or macro-operators [6] possesses. This will allow our learning method to provide search control when the others cannot. Extending the implementation to effect within-domain and cross-domain transfer will also increase its utility.

In conclusion, our study of internal analogy has described a new process for transferring knowledge within a single problem. It has also provided a more complete analysis of the processes needed for analogical transfer than has previously been presented. We specify what constitutes an appropriate source and what knowledge to transfer to the target. In addition, the mechanism is tightly integrated into a general problem solver and does not require an alternate reasoning engine or large case libraries. The psychological validity of the mechanism has also been supported through the psychological data presented.

Acknowledgements This research was supported in part by DARPA contract number F33615-87-C-1499 Amendment 20, and AIP-ONR contract number N00014-86-K-0678-N123. Angela Kennedy Hickman was also partially supported during this research by a Zonta International Amelia Earhart fellowship. We would like to thank Peter Shell for his invaluable contribution to the implementation. Jaime Carbonell, Jill Fain Lehman and David Plaut increased the clarity and the content of the paper through their comments. Sandra Esch and David Plaut assisted in setting up the lab for the experiments.

References

1. Adelson, Beth, "Analogical Reasoning: Problem-Driven Mapping and Debugging," Tech. report, Department of Computer Science, Tufts University, 1989.

2. Anderson, J. R., "A Theory of the Origins of Human Knowledge," *Artificial Intelligence*, Vol. 40, 1989, pp. 313-351.
3. Carbonell, J. G. and Veloso, M. M., "Integrating Derivational Analogy into a General Problem Solving Architecture," *Proceedings of the First Workshop on Case-Based Reasoning*, Morgan Kaufmann, May 1988.
4. Carbonell, J. G., "Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition," in *Machine Learning, An Artificial Intelligence Approach, Volume II*, Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., eds., Morgan Kaufmann, 1986, pp. 371-392.
5. Chi, M. T. H., Feltovich, P. J. and Glaser, R., "Categorization and representation of physics problems by experts and novices," *Cognitive Science*, Vol. 5, 1981, pp. 121-152.
6. Fikes, R. E. and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence*, Vol. 2, 1971, pp. 189-208.
7. Gentner, D., "Structure mapping: A theoretical framework for analogy," *Cognitive Science*, Vol. 7, 1983, pp. 155-170.
8. Gick, M. L. and Holyoak, K. J., "Analogical Problem Solving," *Cognitive Psychology*, Vol. 12, 1980, pp. 306-355.
9. Hickman, A. Kennedy, Shell P. and Carbonell, J. G., "Internal Analogy: Reducing Search during Problem Solving," *Proceedings of AAAI-90*, 1990, (submitted).
10. Jones, R., *A Model of Retrieval in Problem Solving*, PhD dissertation, Department of Information and Computer Science, University of California, Irvine, 1989.
11. Kolodner, J. L., "Maintaining Organization in a Dynamic Long-Term Memory," *Cognitive Science*, Vol. 7, 1983, pp. 243-280.
12. Kolodner, J. L., "Reconstructive Memory: A Computer Model," *Cognitive Science*, Vol. 7, 1983, pp. 281-328.
13. Laird, J. E., Rosenbloom, P. S. and Newell, A., "Chunking in SOAR: The Anatomy of a General Learning Mechanism," *Machine Learning*, Vol. 1, No. 1, 1986, pp. 11-46.
14. Langley, P., Thompson, K., Iba, W. F., Gennari, J. and Allen, J. A., "An Integrated Cognitive Architecture for Autonomous Agents," Tech. report 89-28, Department of Information and Computer Science, University of California, Irvine, 1989.
15. Larkin, J. H., Reif, F., Carbonell, J. G. and Gugliotta, A., "FERMI: A Flexible Expert Reasoner with Multi-Domain Inferencing," *Cognitive Science*, Vol. 12, 1988, pp. 101-138.
16. Lewis, C., "Why and How to Learn Why: Analysis-based Generalization of Procedures," *Cognitive Science*, Vol. 12, 1988, pp. 211-256.
17. Pirolli, P., "A Model of Purpose-driven Analogy and Skill Acquisition in Programming," *The 9th Annual Conference of The Cognitive Science Society*, Lawrence Earlbaum Associates, 1987, pp. 609-621.
18. Reif, F. and Heller, J. I., "Knowledge structure and problem solving in physics," *Educational Psychologist*, Vol. 17, 1982, pp. 102-127.
19. Schank, R. C., "Language and Memory," *Cognitive Science*, Vol. 4, 1980, pp. 243-284.
20. Shell, P. and Carbonell, J. G., "The FRuleKit Reference Manual", CMU Computer Science Department internal paper.
21. Simon, D. P. and Simon, H. A., *Individual Differences in Solving Physics Problems*, Lawrence Earlbaum Associates, Hillsdale, New Jersey, 1978, ch. 13.
22. Veloso, M. M. and Carbonell, J. G., "Learning Analogies by Analogy - The Closed Loop of Memory Organization and Problem Solving," *Proceedings of the Second Workshop on Case-Based Reasoning*, Morgan Kaufmann, May 1989.