Exploring the Impact of Data-driven Tutoring Methods on Students' Demonstrative Knowledge in Logic Problem Solving

Behrooz Mostafavi
North Carolina State University
Raleigh, NC 27695
bzmostaf@ncsu.edu

Tiffany Barnes
North Carolina State University
Raleigh, NC 27695
tmbarnes@ncsu.edu

ABSTRACT

We have been incrementally adding data-driven methods into the Deep Thought logic tutor for the purpose of creating a fully data-driven intelligent tutoring system. Our previous research has shown that the addition of data-driven hints, worked examples, and problem assignment can improve student performance and retention in the tutor. In this study, we investigate how the addition of these methods affects students' demonstrative knowledge of logic proof solving using their post-tutor examination scores. We have used data collected from three test conditions with different combinations of our data-driven additions to determine which methods are most beneficial to students who demonstrate higher or lower knowledge of the subject matter. Our results show that students who are assigned problems based on profiling proficiency compared to prior exemplary students with similar problem-solving behavior show higher examination scores overall, and the use of proficiency profiling increases retention and reduces the amount of time taken in-tutor for lower performing students in particular. The results from this study also helps differentiate the behavior of higher and lower performing students in tutor, which can allow quicker interventions for lower proficiency students.

Keywords

Data-driven Methods, Proficiency Profiling, Tutoring Systems

1. INTRODUCTION

We have been incrementally adding data-driven methods for problem assignment[9, 10], hint generation[3], and worked examples[11] to the Deep Thought logic tutor to create a fully data-driven tutoring system. While we have observed improvements in student retention and tutor scores with each of these additions, we have not studied the difference in post-tutor examinations when these methods are combined in different test conditions. We seek to understand how the specific methods of problem assignment and combination of hints and worked examples may have impacted student performance on related questions on the course midterm exam.

In this paper we compare two classrooms of students using different test conditions of Deep Thought, with different combinations of problem assignment, hints and worked examples. Students' knowledge of logic were evaluated in two problems on a mid-term exam, and these scores were used to differentiate high and low proficiency students for our analysis. The results from our analysis show that high performing students benefit most from problem-solving opportunities, while low performing students benefit most from problem assignment based on proficiency profiling, comparing current students to prior exemplary students with similar behavior. We conclude that the use of proficiency profiling is the most effective method for increasing retention and reducing time spent in the Deep Thought tutor, and result in higher overall examination scores. The results from this study also help differentiate the behavior of higher and lower performing students in tutor, allowing for quicker interventions for lower proficiency students who need additional instructional support.

2. RELATED WORK

Koedinger et al.[6] summarized the general process of intelligent tutoring systems: the system selects an activity for the student, evaluates each student action, suggest a course of action (either via hints, worked examples, or another form of feedback), and finally updates the system's evaluation of the student's skills. An effective tutor should adapt instruction according to the student's current knowledge level [1]. However, in order to make instructional decisions, most ITSs either use fixed pedagogical policies providing little adaptability, or expert-authored pedagogical rules based on existing instructional practices [1, 14]. Intelligent tutoring systems with data-driven methods can be more adaptive by leveraging previous student data in order to complete one or more of these steps. Data-driven approaches to making effective pedagogical decisions – in particular selecting problems, when to apply worked examples, and the type of hint or feedback to provide – would mostly bypass the need for expert involvement in creating and improving the effectiveness of ITSs. In practice, incorporating student data has been shown to increase learning efficiency and predict student behavior. This, in particular is why we use data-driven knowledge tracing (DKT) of rule applications within the Deep Thought logic tutor to facilitate profiling of students' proficiency.

In the remainder of this section, we describe the Deep Thought logic tutor and the data-driven additions implemented. We then describe the system and data used to evaluate the effectiveness of these data-driven methods in Deep Thought. After reporting the results of this evaluation, we discuss the implications for future design decisions in the tutor, and present our conclusions.

2.1 The Deep Thought Tutor

We have been examining the potential for data-driven methods to improve learning gains in a complex problem solving domain by incrementally augmenting the Deep Thought logic tutor. Deep Thought is a tutor for graphically constructing propositional logic proofs. Deep Thought presents proof problems consisting of logical premises and a conclusion to be derived using logical axioms. Deep Thought is divided into 6 levels of logic proof problems. In previous work with the Deep Thought logic tutor, we have been implementing data-driven methods for several of the intelligent tutor steps. We implemented a data-driven mastery learning system (DDML) to track student actions and assign appropriate problems based on the student's current level of proficiency [9]. The problem set was split into two tracks: a high proficiency track and a low proficiency track for Levels 2–6, with Level 1 containing a common set of problems for initial track assignment. We tracked student actions throughout their time in the tutor, and in particular their application of logical rules to construct logic proofs. Based on their correct or incorrect application of logical rules, the DDML updated a set of rule scores, one score for each logical rule. At the end of each level, the students' rule scores were weighted based on expert-determined priorities; rules deemed by experts to be of high importance to solving the problems in that level were weighted higher than rules that were not. These weighted scores were summed together, and compared to the average rule scores in the previous semester's data; based on this comparison, students were assigned to the higher or lower proficiency path. We tested Deep Thought with the DDML incorporated and found students completed, on average, 79% of all six levels in the tutor assignment. Student retention rate was 55%. This was an improvement over the non-DDML version of Deep Thought (61% tutor completion on average, and 31% retention rate).

We later incorporated a data-driven proficiency profiler (the DDPP) to replace the expert-determined priorities[10][8]. The DDPP is a system that calculates student proficiency at the end of each level in Deep Thought based on how a given student performs in comparison to exemplars who employed similar problem solving strategies, with rule scores weighted as determined through principal component analysis (PCA). Based on how similar exemplary students were assigned in subsequent levels, the DDPP can determine the best proficiency level for a new student. In contrast to the DDML system previously employed, this proficiency calculation and rule weighting is entirely data-driven, with no expert involvement.

We determined similar problem solving strategies among the exemplars by clustering the exemplars' rule scores based on hierarchical clustering. Expert weighting was replaced by PCA of the frequency of the rules used for each exemplar for each level, accounting for 95% variance of the results. For each rule, its PCA coefficient is the new weight for that rule score. When a new student uses the tutor, the student's rule scores are calculated throughout the level. At the end of each level, the DDPP examines each student's individual rule score and assigns it to a cluster for that rule. The DDPP then finds which clusters the scores for the most important rules fall into for that level (based on the same PCA based weighting), and then classifies that student into a *type* based on the set of clusters the student matches. Finally the system assigns the student to a proficiency track based on data from the matching type of exemplars, and how those exemplars were placed in the next level. The more exemplars we have of a given type, the stronger the prediction we can make for a new student. In the event that a new student doesn't match an existing type in the exemplar data, the student's proficiency is calculated using the average scores, as in the original DDML system.

Providing hints to students in the course of an intelligent tutor as a possible form of step-based feedback has the potential to increase learning gains. Razzaq, Leena, and Heffernan [12] found that learning gains increased for students given on-demand hints in comparison to students who were provided hints proactively. In Deep Thought, the hint system used is called Hint Factory. Hint Factory is an automatic data-driven hint generator that converts an interaction network graph of student trace behavior into a Markov decision process (MDP) to automatically select on-demand hints for students upon request, based on their individual performance on specific problems. The MDP is data-driven, using actions logs from previous Deep Thought use in the classroom to assign weight to proof-state actions based on whether or not that action ultimately led to successful completion of the proof. These hints help students solve problems by suggesting what step should be taken next on a multi-step problem. Hint Factory has been implemented in the Deep Thought logic tutor to automatically deliver context-specific hints to students during problem-solving [4]. In a previous study Hint Factory was shown to provide context-specific hints over 80% of the time [3]. In a pilot study, Barnes & Stamper found that Hint Factory can provide sufficient, correct, and appropriate hints for the Deep Thought Logic tutor and help students to solve more logic proof problems in the same span of time [4]. However, we currently cannot determine the effect hints would have in addition to the DDML or DDPP; so far, students using either of those versions of Deep Thought did not use hints often enough for any meaningful analysis.

Adding worked examples as a supplement to traditional problem solving can also be beneficial [2, 13]. Hilbert and Renkl [5] found that improved learning outcomes occurred when providing worked examples with a prompt, and proposed that this was due to allowing the students to have a greater cognitive load at once. McLaren and Isotani [7] compared three tutors using all worked examples, all traditional unguided problem solving, and a mix of worked examples and problem solving. Each group achieved similar learning gains, but the students who were given all worked examples required less time to achieve those gains. We added worked examples to the version of Deep Thought with the DDML incorporated[11]. Worked examples were generated based on previous best student solutions, and procedurally annotated. They were presented to students randomly on a per-problem basis, based on the number of problems they had solved in that level already. We found that student retention overall was 90%, and students completed 94% of the tutor on average. This percentage was significantly higher than that of the DDML alone.

3. METHODS

Deep Thought was used as a mandatory homework assignment by students in an undergraduate "discrete mathematics for computer scientists" course in Fall 2015 and Spring 2016. Students in the two semesters were taught by different instructors. Students were assigned Levels 1–6 of Deep Thought for full credit, with partial credit awarded proportional to the number of levels completed. For this study, we compare the data from three Deep Thought test conditions used across the two semesters to differentiate the effect of our data-driven methods on student performance.

The first group evaluated for this study were assigned only problem-solving opportunities (PS group, $n = 26$). The problem assignment system used was the DDML system described in the previous section, where students were assessed between levels and placed on either a high or low proficiency track in the next level. This group of students were taken only from the Fall 2015 semester, as there existed no equivalent test condition in Spring 2016.

The second group of students were randomly assigned either problem-solving opportunities or worked examples of the same problems within each level (PS/WE group, $n = 179$), with the number of problem-solving opportunities controlled to match the number of problems solved by the PS group. Like the PS group, the PS/WE group were assigned proficiency tracks using the DDML. However, because individual rule application scores were updated at each step in worked examples as if a student had applied that rule in while problem solving, most students were consistently assigned to the high track in most levels, and were only assigned the low track when their individual performance was below satisfactory. This group of students were taken from both the Fall 2015 and Spring 2016 semesters.

The third group of students were randomly assigned problem-solving opportunities or worked examples in the same manner as the PS/WE group, but with the DDPP method assigning proficiency tracks instead of the DDML, where students were assigned the same proficiency track as prior students who most closely matched their rule application behavior (DDPP group, $n = 61$). This group of students were also taken from both the Fall 2015 and Spring 2016 semesters. Students in all three groups had access to on-demand hints.

All students were evaluated using two proof problem questions as part of a mid-term examination, which was used as a post-test for this study. Students performance in the post-test for both Fall 2015 and Spring 2016 were graded by the same teaching assistant, ensuring consistent evaluation across all results. Students were separated for evaluation by performance on the post-test and by the predominant track level in Deep Thought. The post-test was a set of two proofs students had to solve on paper for a midterm exam. These questions were hand-graded with partial credit given based on the percentage of the proofs completed and points taken off for misapplication of rules and skipping non-trivial rules. We considered two performance levels: post-test scores greater than or equal to 80% (AB), or less than 80% (CDF). The post-test scores mark the final evaluation of students' ability to solve proof problems, and occurs immediately following the Deep Thought tutor homework assignment.

The second dimension we studied was the proportion of high to low proficiency track levels the students completed. Students who were assigned to the high proficiency track in a level had the ability to finish on either the high or low proficiency track depending on the number of problems skipped within that level. Students who completed more levels on the high track than the low track were marked as high track students, and students who completed more levels on the low track than the high track were marked as low track students. The track assignments indicate the number and complexity of problems students received, with the low track having more problems of lower complexity, and the high track having fewer problems of higher complexity. The tracks were designed so that students would have a similar number of rule applications across the tracks, even though the number of problems differs. Typically, the low track has three problems with expert solutions using 5 rule applications, and the high track has 2 problems with expert solutions using 7 – 8 rule applications - meaning that both tracks minimally required about 15 total rule applications (though students typically used more).

In addition to post-test and predominant track level, we examined total time in tutor, average time spent per problem, percentage of correct rule applications out of all rule applications, and the total number of rule applications. We also looked at ancillary behaviors (hint usage, skipped problems, and reference requests) that could differentiate high and low performing students. We compared these metrics to better understand the impact of worked examples, hints, and data-driven track selection on student performance. The results of this descriptive analysis are presented in the next section.

4. RESULTS

Table 1 displays the percentage of AB students in each of the PS, PS/WE, and DDPP groups for all students, as well as students who completed the majority of the tutor in either the high or low tracks. Table 1 also displays the percentage of students in each group and each track who dropped out of the tutor before full completion, as this is one of the metrics we have used to judge the effectiveness of our data-driven methods. In our previous work using the same version of Deep Thought, we found that students completed 94% of the tutor on average, with a retention rate of 90%. The average percent tutor completion for the groups in this study were consistent with these numbers (PS: 95%, PS/WE: 93%, DDPP: 94%).

The first interesting result of note is that the percentage of students who performed better on the post-test was higher

Table 1: Percentage of AB Students and Percentage of dropped students in the PS, PS/WE, and DDPP groups.

Condition		ALL	High Track	Low Track
	n	*% AB Students*		
PS	*26*	65.38	71.43	63.16
PS/WE	*179*	49.72	52.35	36.67
DDPP	*61*	63.93	66.67	61.76
	n	*% Dropped Students*		
PS	*26*	3.85	0.00	5.26
PS/WE	*179*	11.73	6.71	36.67
DDPP	*61*	9.84	11.11	8.82

for for the PS (65%) and DDPP (64%) groups than for the PS/WE group (50%), across all the students, as well as within the high and low track groups. In the PS group, students who completed more levels on the high track displayed a higher overall proficiency of the subject matter than those who finished more often on the low track (71% vs 63%, respectively), as did students in the PS/WE group (52% vs 37%).

However, students in the DDPP group showed a consistent level of proficiency regardless of the tracks completed (66% vs 61%), which makes sense considering that these students were matched to previous successful students who displayed similar rule-application behavior, and had a more even placement within the high and low tracks compared to the PS group, who had even placement among tracks, but within the context of their own performance compared to expert-decided thresholds. The DDPP group also had higher placement compared to the PS/WE group, who were placed on the high track much more often than not due to the inclusion of worked examples. A Kruskal-Wallis test for one-way analysis of variance showed no significant difference between groups ($p = 0.22$).

Students also had a higher retention rate in both the PS (4%) and DDPP (10%) groups compared to the PS/WE group (12%). It is especially interesting to see the drop rate among low track students in the PS/WE group, who had a much lower retention rate among all the students in the study. Because students in the PS/WE group were more often that not placed in the high track in each level, for students to end up on the low track indicates a high level of problem-skipping among these students. We can conclude that low performing students who are not intelligently assigned problems based on their problem-solving performance appear to gain little from worked examples.

While it may be tempting to declare problem-solving opportunities with no worked examples as the best performing pedagogical choice among the three groups based on these numbers alone, a look into additional performance metrics gives some more insight. Table 2 presents the amount of time spent in tutor and on each problem, as well as the percentage of correct and total rule applications for each group, separated by track. The numbers presented are the median values for each metric, since the distributions of scores were highly skewed and non-normal, and none of the differences were significant due to low sample size within each subgroup.

As shown in Table 2, among AB students in all three groups, the total time spent in tutor appears similar, although the mean time for high-track students was lower for DDPP (M = 3.95hr, SD = 6.21hr) compared to PS/WE ($M = 4.46hr$, $SD = 9.13hr$) and PS ($M = 6.66hr$, $SD = 9.91hr$). The mean time for low-track students was lower for PS ($M = 4.63hr$, $SD = 9.55hr$) and DDPP ($M = 5.48hr$, $SD = 5.42hr$) than the PS/WE ($M = 7.74$, $SD = 9.76$). The means of average problem time, percentage of correct rule applications, and number of rule applications were consistent with the median values presented in Table 2 across all three groups. Note that low-track students in the PS/WE groups had the lowest percentage of correct rule applications, and the highest number of total rule applications among all the groups. This means they are doing more work, but a lower percentage of it is correct.

As shown in Table 2, among CDF students in all three groups, the total time spent in tutor is dramatically different, with PS spending 3 to 4 times as long in the tutor than PS/WE and DDPP groups. This ratio is also similar in the average problem time for high and low track students, and the number of total rule applications for high track students. Therefore, while problem-solving only (PS) may have a slightly higher overall success rate in helping students learn proof problem solving and remain in the tutor than the DDPP students, for students who are less prepared, PS results in a much higher time spent in the tutor, with little return on the time investment. Therefore, for students who have a better grasp of the subject matter, pure problem-solving may offer a slightly better option for getting through the assigned tutor, although the differences between problem solving, problem solving and worked examples, and proficiency profiled assigned problem solving and worked examples are minimal. However, for less prepared students, pure problem-solving opportunities offer little to guide students to higher understanding of the material, and in general, the DDPP offers a much better path to completing the tutor in far less time for both AB and CDF students, giving students the opportunity to encounter all the subject matter and have a greater chance of learning the material, resulting in higher overall post-test scores.

Completing the tutor assignment is important for students; however, since we want to make sure that students are learning the material well, mid-term examination scores are ultimately a higher gauge for learning success. Among all the experimental groups in this study, at most 65% of students were performing at A or B grade level on the mid-term examination. We would like to increase this percentage of AB students, so the question at this point is: Is it possible for us to predict low exam scores based on in-tutor data for early intervention?

We first look at the differences between AB and CDF students in Table 2, with the assumption that the DDPP method offers the best overall chance of success for students. For high track students, total tutor time, average problem time, percentage of correct rule applications, and total rule applications are consistent between AB and CDF students. However, for low track students, average problem time, percentage of correct rule applications, and total rule applications show a higher difference. CDF students spent twice as

Table 2: Total Time, Average Problem Time, Percentage of Correct Rule Applications, and Total Rule Applications for AB and CDF students in the PS, PS/WE, and DDPP groups, separated by High and Low Track. The numbers listed are all median values.

		AB STUDENTS				CDF STUDENTS		
		PS	**PS/WE**	**DDPP**		**PS**	**PS/WE**	**DDPP**
HIGH TRACK	***n***	*5*	*78*	*18*	***n***	*2*	*71*	*9*
Total Tutor Time (hr)		2.47	2.37	2.80		12.8	3.75	3.17
Average Problem Time (min)		9.89	11.1	12.1		52.3	18.4	16.0
% Correct Rule Applications		60.8	63.5	58.5		64.1	56.9	62.3
Total Rule Applications		258	214	203		471	255	204
LOW TRACK	***n***	*12*	*11*	*21*	***n***	*7*	*19*	*13*
Total Tutor Time (hr)		1.80	3.33	3.67		17.2	5.96	4.98
Average Problem Time (min)		6.76	15.2	15.0		60.1	25.0	30.4
% Correct Rule Applications		68.8	45.5	57.0		48.7	45.7	47.0
Total Rule Applications		201	404	291		382	394	389

long on average per problem than AB students, and applied rules correctly less than half of the time, while AB students applied rules more than half of the time. CDF students also attempted applying rules 25% more overall than AB students.

Since the performance differences between AB and CDF students are not as apparent for high track students, we look at ancillary tutor behavior to make a better distinction. Table 3 shows the number of requested hints, the number of skipped problems, and the number of rule reference requests (descriptions of logic rule operations) made by students in all groups. For the DDPP group, the most apparent difference among AB and CDF students are the number of hints requested, with the CDF group requesting 32 hints ($M = 50$, $SD = 57$) compared to 17 ($M = 32$, $SD = 42$) for the AB group. This difference in hints requested between AB and CDF students is also consistent across all groups and both high and low track students. We conclude that for high track students, we can differentiate between higher and lower proficiency students using hint request behavior, and for low track students, we can differentiate higher and lower proficiency students using the amount of time spent on average per problem and the percentage of correct rule applications. This allows the possibility of making an intervention during a student's progress through Deep Thought in the case that a student requires additional feedback or aid from an instructor due to a lesser understanding of the subject matter.

Table 3: Number of Hints, number of Skips, and number of Rule Reference requests for AB and CDF students in the PS, PS/WE, and DDPP groups, separated by High and Low Track. The numbers listed are all median values.

	PS		PS/WE		DDPP	
HIGH	**AB**	**CDF**	**AB**	**CDF**	**AB**	**CDF**
# Hint	95	166	12	26	17	32
# Skip	5	16	1	1	0	2
# Ref	151	168	76	145	111	92
LOW	**AB**	**CDF**	**AB**	**CDF**	**AB**	**CDF**
# Hint	30	104	31	44	19	26
#Skip	1	0	30	24	3	15
# Ref	77	224	60	271	55	109

5. CONCLUSION

In this paper we compared two classrooms of students using different test conditions of Deep Thought, with different combinations of problem assignment (DDML or DDPP) and the addition of worked examples, for the purpose of understanding how the specific methods of problem assignment and combination of hints and worked examples affect high and low performing students, as evaluated using mid-term examination scores. We found that for higher proficiency students who have a firmer grasp of the subject matter, problem-solving opportunities offer the best chance of completing the tutor in a timely manner; however, the addition of worked examples does not significantly detract from these students' learning experience. The method of problem assignment (DDML or DDPP) does not have a noteworthy effect on high student performance.

For lower proficiency students, we found that problem-solving opportunities alone with DDML problem assignment offered little to guide students to higher understanding of the material, and greatly extended the amount of time students spent in the tutor with little learning benefit. The addition of worked examples helped these students get through the tutor faster, however these students had a lower retention rate than any other students and lower examination scores. We conclude from these results that updating our data-driven skill estimates equally for viewing or applying rules resulted in students being assigned to the high-track when they were not prepared to solve harder problems. With proficiency profiling – matching students to previously successful students and the paths they take through the tutor – we can reduce the amount of time spent in tutor, increase retention, and make better use of worked examples by giving them alongside problems that better match an individual student's proficiency level. This results in similar performance to problem solving alone in terms of retention and knowledge gained, but with a lot less time spent in the tutor for lower-proficiency students. We conclude that our DDPP method offers the best overall possibility of success for students completing the Deep Thought tutor in a timely manner, learning the subject matter, and performing well on post-tutor examinations.

6. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grants 1432156 and 0845997.

7. REFERENCES

[1] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive Tutors: Lessons Learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[2] R. K. Atkinson, S. J. Derry, A. Renkl, and D. Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, 70(2):181–214, 2000.

[3] T. Barnes, J. Stamper, L. Lehmann, and M. J. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197 – 201, 2008.

[4] T. Barnes, J. Stamper, L. Lehmann, and M. J. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197 – 201, 2008.

[5] T. S. Hilbert and A. Renkl. Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior*, 25(2):267–274, 2009.

[6] K. R. Koedinger. New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization. *AI Magazine*, 34(3):27–41, 2013.

[7] B. M. McLaren and S. Isotani. When is it best to learn with all worked examples? In *Artificial Intelligence in Education*, pages 222–229, 2011.

[8] B. Mostafavi and T. Barnes. Data-driven Proficiency Profiling - Proof of Concept. In *Proceedings of the 6th International Conference on Learning Analytics and Knowledge (LAK 2016). In Press.*, 2016.

[9] B. Mostafavi, M. Eagle, and T. Barnes. Towards data-driven mastery learning. In *Proceedings of the 5th International Conference on Learning Analytics and Knowledge (LAK 2015)*, pages 270–274, 2015.

[10] B. Mostafavi, Z. Liu, and T. Barnes. Data-driven Proficiency Profiling. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, pages 335–341, 2015.

[11] B. Mostafavi, G. Zhou, C. Lynch, M. Chi, and T. Barnes. Data-driven Worked Examples Improve Retention and Completion in a Logic Tutor. In *Proceedings of the 17th International Conference on Artificial Intelligence in Education (AIED 2015)*, pages 726–729, 2015.

[12] L. Razzaq and N. T. Heffernan. Hints: is it better to give or wait to be asked? In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*, pages 349–358. Springer, 2010.

[13] J. Sweller and G. A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):59–89, 1985.

[14] K. VanLehn. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16(2):227–265, 2006.