Determining Problem Selection for a Logic Proof Tutor

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

When developing an intelligent tutoring system, it is necessary to have a significant number of highly varied problems that adapt to a student's individual learning style. In developing an intelligent tutor for logic proof construction, selecting problems for individual students that effectively aid their progress can be difficult, since logic proofs require knowledge of a number of concepts and problem solving abilities. The level of variation in the problems needed to satisfy all possibilities would require an infeasible number of problems to develop. Using a proof construction tool called Deep Thought, we have developed a system which chooses existing problem sets for students using knowledge tracing of students' accumulated application of logic proof solving concepts and are running a pilot study to determine the system's effectiveness. Our ultimate goal is to use what is learned from this study to be able to automatically generate logic proof problems for students that fit their individual learning style, and aid in the mastery of proof construction concepts.

Keywords

Logic Proof, Problem Selection, Knowledge Tracing, Intelligent Tutor.

1. INTRODUCTION

Logic proof construction is an important skill in several fields, including computer science, philosophy, and mathematics. However, proof construction can be difficult for students to learn, since it requires a satisfactory knowledge of logical operations and their application, as well as strategies for problem solving. These required skills make developing an intelligent tutor for logic proof construction challenging, since a number of variables must be taken into account when selecting problems for students that promote learning of proof concepts that fit their individual learning styles.

We describe the on-going development of an intelligent tutor, and an initial experiment to determine the effectiveness of knowledge tracing methods used to select sets of problems for students. For the study, we have built upon an existing, non-intelligent proof construction tool called Deep Thought, which has previously been used for proof construction assignments, and from which student performance data has been collected.

Our long-term goal is to provide a system for logic proof construction that adapts to a student's individual learning abilities, using that student's previous performance in logic rule application and problem solving in order to automatically generate problems that aid in mastery of core proof construction concepts. It is also our goal to develop the system in such a manner that it is domain independent, and can be applied to other fields that have multiple concepts and skills that need to be demonstrated.

2. THE DEEP THOUGHT TUTOR

2.1 The Original System

Deep Thought is a web-based proof tool with a graphical user interface that provides a set of problems that display logical premises, buttons for logical rules, and a conclusion that a student must prove by applying those rules to the premises (Figure 1). Deep Thought was originally developed as a practice tool and system for proof construction assignments. In its original form, Deep Thought provides students with three levels of problems, with problems in each level requiring a different set of logical rules for completion (Level 1: Inference rules; Level 2: Inference rules [more difficult]; Level 3: Inference and Replacement rules). Problems are selected from a drop-down menu, and students can select and complete problems in any order. As a student works through a problem, each step is logged in a data file that records a number of attributes, including the current problem, the rule being applied, any errors made (such as attempting to use a rule that is logically impossible), completion of the problem, time taken per step, and elapsed time taken to solve the problem.

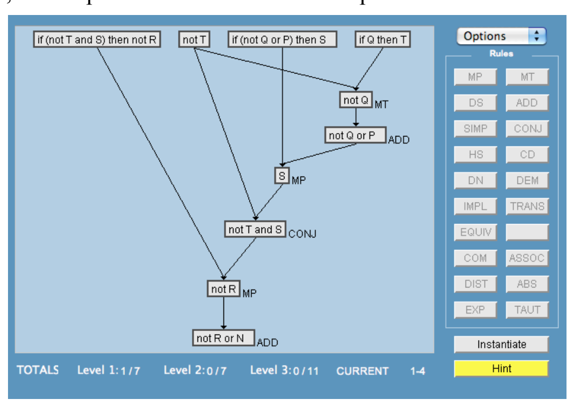

Figure 1. The Deep Thought user interface.

2.2 The New System

A number of changes were made to Deep Thought in order to create an intelligent system. Notable changes important to this study are described below.

2.2.1 Problem Set

Instead of allowing students to select problems at will, the new system provides an ordered set of problems for students to solve. Problem selection is determined based on the level of rule application and difficulty students are expected to demonstrate. Students can skip problems within the current level; however, they must complete all problems within that level to proceed to the next.

The original problem set for Deep Thought was expanded to give a wide variety of problems while maintaining the rule applications required and difficulty level of the original set. These changes were made and tested by domain experts to ensure consistency between the old and new system for performance comparison. The problem set was changed as follows:

- Levels 1 & 2: Inference rules
- Levels 3 & 4: Inference rules [more difficult]
- Levels 5 & 6: Inference and Replacement rules
- Level 7: Inference and Replacement rules [more difficult, not present in original set]

Level 1 contains 3 problems common to all users. With no prior performance data available, these 3 problems serve the purpose of collecting initial performance data to select problems in the next level.

Levels 2 – 6 are split into two difficulty tracks (easy and hard), to which students are sent based on their prior performance. Both tracks within each level contain problems that require similar rule applications and proof concept demonstration. The hard path contains 2 problems and the easy path contains 3 problems, each with an alternate problem (the alternate problem contains the same number of steps and same rules as the original, but with different ordering of required rule applications). The difficulty of problem sets were determined by domain experts who have many years of experience working with the types of proofs presented in Deep Thought and with students working through those problems.

Level 7 contains 3 problems common to all users. These problems were not present in the original set, but were added to test student skills obtained by working through the rest of the tutor. The problems in this level were more difficult than any other problems in Deep Thought.

2.2.2 *Problem Selection*

Problem selection in Deep Thought is determined using two methods. The first is the decision process that occurs between levels that sends a student down difficulty paths. The second is the process that selects problems within the current level.

For the difficulty path decision process, data from a student's work in Deep Thought is recorded and used to update a set of action scores. The scores for each rule are given an initial value, and are then updated based on the actions taken by the user, with correct applications of rules increasing the rule score, and incorrect actions (errors) decreasing them. The calculations for rule updates are made using a Bayesian knowledge-tracing model [2].

At the end of each level, the scores for each action are compared to average scores from historical student data collected using the old version of Deep Thought. The scores from the old version were calculated using the same bayesian knowledge tracing model after students had worked through the existing problems sets, and were used as a threshold value. Each rule is given a value of 1 if the score is higher than the threshold and given a value of -1 if the score is lower than the threshold. For each action, these values are weighted based on the rule priority for each level (primary or secondary), and then summed. A sum less than zero sends the students down the easy path, and a sum greater than zero send the students down the hard path.

Within each level, problems are selected using a decision tree process, based on whether or not students skip problems. Students who are working within the easy difficulty track are given the alternate problem if they choose to skip the original problem presented, with the idea that the difference in rule application order can allow them to approach the concept in a different manner. For students working in the hard difficulty track, skipping more than the first problem in the set will send them to the easy difficulty track. If students solved one problem in the hard difficulty path before being sent to the easy difficulty path, they are not required to solve the corresponding problem in the easy path, in order to maintain the number of problems required to complete the level.

The reason for the skipped problem decision process is to compensate for students who may have shown proficiency in a previous level, but have a harder time solving the next set of problems. Students who have been sent down the hard difficulty track are expected to have satisfactory mastery of concepts needed for the next set of problems, without the need for alternate problems. If students have difficulties with the harder set, they are given the opportunity to work through a greater number of easier problems in order to practice those concepts required before moving on to the next level.

3. METHOD & INITIAL RESULTS

The new system of Deep Thought was used as an assignment in two sections of a Computer Science Logic & Algorithms class taught by the same instructor. Deep Thought was run as a web applet, with students allowed to work through the problem sets at their own pace. The more difficult Level 7 problems were made optional to students, as they were not presented in the original curriculum for the course. Student data was recorded in two separate tables in a database stored on a server which communicated with the Deep Thought applet. The two tables used were:

- Log Table: This table was used to track information specific to individual students, including information used for tracking a student's progress in the system (log-in information, current working level / difficulty / problem, skipped and completed problems in the current level, levels completed and at which difficulty track) as well as data used for the knowledge tracing process (updated scores for individual rules and concepts).
- Data Table: This table was used, as in the original system, to track each action taken by students while solving proofs for analysis (level / difficulty / problem, the rule being applied, errors made, screen state, hints used, action step time, and total elapsed time for the current problem).

A total of 63 students worked through the new version of Deep Thought. Of these students, 32 completed at least through Level 6 of the problem sets, with the majority of drop-outs occurring after Level 4. The number of students who did not complete Deep Thought was high (over 50%), however it should be noted that the professor for the class used for the experiment had not completely covered Replacement rules (used in Level 5 onwards) at the time these results were reported. A flow diagram showing the path students travelled while using Deep Thought is shown in Figure 2.

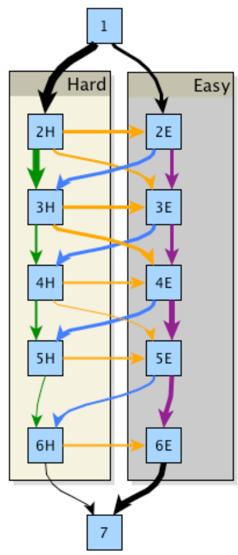

Figure 2: Flow diagram of student path through Deep Thought problem sets. The thickness of the arrows is weighted based on the number of students travelling that path.

Based on the diagram in Figure 2, the following conclusions can be drawn regarding the paths commonly taken by the students. Most of the class was able to complete the hard paths for levels 1 and 2, with most of the students being sent down the hard path once level 1 was completed and staying through level 2. At level 3 however, some of the students were sent to the easy path, either at the same level or at level 4. From level 4 onwards most of the class stayed on the easy path (those who completed Deep Thought). From Level 5 onwards, most of the students stayed on the easy path until completion.

Based on the system and our goals for it, these paths are what would be expected. The problems at levels 1 and 2 are basic inference problems, and are designed to be easier to solve for students with the expected requisite knowledge. Level 3 was where the problems were designed to increase in difficulty. Students should not have been able to complete level 3 without showing a higher level of proficiency than had been required up until that point if the problem selection was effective. The fact that most of the class was transferred to the easy path at level 3 indicates that this is the case; students were given problems that were difficult enough to challenge them on the hard path (to the point of being sent to the easy path at the next level) while still being manageable on the easy path.

Since most students did not complete Deep Thought past this point, the paths from level 4 on are somewhat skewed. However, the fact that the students who did complete Deep Thought through level 7 remained on the easy path indicates that the problems in levels 4, 5, and 6 were overall appropriately difficult. These problems were meant to be challenging regardless of the path the student was on, particularly considering that the students did not have requisite knowledge of replacement rules at this point. Therefore the fact that most students stayed on the easy path through level 7 indicates that the problems given were at an expected level of difficulty for them. Conversely, if more students had been able to stay on, or move to the hard path at these levels, it would indicate that either the system is selecting problems that are too easy, or the problems themselves were not designed to be challenging enough. Since the students were continually put on the easy path at these levels, neither of these situations is the case.

4. FUTURE WORK

The data from this initial experiment needs further analysis before any new features are added to the system. However, initial results are promising, and it appears that the system is effective in selecting problem sets for students at a general level. Once this data has been analyzed further and compared to previous data from the old version of Deep Thought, we can make more definite assumptions about the effectiveness of our problem selection.

The next step is to apply the system within levels to test specific problem selection based on rule scores and rule ordering, rather than just problem sets. If that proves effective, we can apply methods in development for automatic generation of problems based on individual rule component construction. Overall, we plan to continue development of Deep Thought into a more effective intelligent tutor in logic proof construction.

5. REFERENCES

[1] Barnes, T. and Stamper, J. 2007. Toward the Extraction of Production Rules for Solving Logic Proofs. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education, Educational Data Mining Workshop (AIED 2007)*, 11-20

[2] Corbett, A.T. and Anderson, J. R. 1994. Knowledge Tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 3, 253-278

[3] Croy, M., Barnes, T., and Stamper, J. 2008. Towards an Intelligent Tutoring System for Propositional Proof Construction. In Briggle, A., Waelbers, K., and Brey, E. (Eds.) *Current Issues in Computing and Philosophy*, 145-155 IOS Press, Amsterdam, Netherlands

[4] Eagle, M., Johnson, M., and Barnes, T. 2012. Interaction Networks: Generating High Level Hints Based on Network Community Clusterings. In *Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012)*, 164-167

[5] Mostafavi, B., Barnes, T., Croy, M. 2011. Automatic Generation of Proof Problems in Deductive Logic. In *Proceedings of the 4th International Conference on Educational Data Mining (EDM 2011)*, 289-294

[6] Murray, T. 1999. Authoring Intelligent Tutoring Systems. In *International Journal of Artificial Intelligence in Education*, 10, 98-129