Feedback Tools and Motivation to Persist in Intro CS Theory

Ivona Bezáková
Rochester Institute of Technology

Kimberly Fluet
University of Rochester

Edith Hemaspaandra
Rochester Institute of Technology

Hannah Miller
Rochester Institute of Technology

David E. Narváez
University of Rochester

ABSTRACT

Introductory assignments in CS Theory ask students to construct instances of various computational models (such as finite automata, regular expressions, context-free grammars, or push-down automata) for a given language. Verifying the correctness of their model instance is challenging for beginner CS Theory students since the concepts are abstract and there are infinitely many possible inputs. The popular JFLAP software allows students to visualize the running of their instance on a specific input. We recently developed a server extension to JFLAP which checks whether a student's instance is equivalent to the instructor's solution and, if not, it returns a "witness string," an input string on which the student's construction and the correct solution differ.

We report on comparing student performance and perceptions in three course sections of CS Theory that differed in the tools the students used for model-construction assignments: no-JFLAP, vanilla JFLAP, and JFLAP+server (with immediate correctness checking). In Spring 2022, 59 out of 86 students consented to include their data in our research. We compared the performance of the three sections on 28 questions (7 DFA problems, 2 NFA problems, 11 regex problems, 6 CFG problems, and 2 PDA problems). We found that while the grades were comparable among the sections, the section using JFLAP+server credited the instantaneous witness string feedback with their increased understanding of the concepts.

1 BACKGROUND

JFLAP [2] visualizes how a student's computational model instance processes a specific input, but it does not indicate whether the instance correctly processes an entire language. Our server extension adds this equivalence checking to JFLAP. In a previous study [1], we found that with only witness string feedback, students keep resubmitting attempts until correct. However, the study was limited to only five homework problems, one per each computational model, and did not study the effects of using just JFLAP (without correctness verification). This work extends the study with significantly more problems (28 in total) and adds a comparison with JFLAP-only. In our setup, the no-JFLAP section did not have any exposure to JFLAP, the JFLAP-only section had to create their homework submissions using vanilla JFLAP (no feedback from the server), and the JFLAP+server section was required to use our server extension (getting the witness string feedback). We collected data on student demographics, grades, Likert scale responses to survey questions, and free text responses. We asked the JFLAP-only and JFLAP+server sections about their perceptions of the software (JFLAP/JFLAP+server) they used, whether it helped them to solve the homework problems, and whether it helped them to understand the homework problems. We asked the JFLAP+server section whether and why they resubmitted to the server; whether they think that allowing unlimited resubmissions until correct is a good way to learn the concepts; and if a binary correct/incorrect grading system should be used.

2 RESULTS AND DISCUSSION

Across the three sections, we found that the median grades for the targeted homeworks were comparable: no-JFLAP had 22 students with median grade of $(95\%{\pm}6\%)$, JFLAP-only had 22 students with median grade of $(90\% \pm 7\%)$, and JFLAP+server had 21 students with median grade of $(96\% \pm 6\%)$.

Students in the JFLAP+server section (the only section with immediate equivalence checking feedback) agreed that the software helped them to solve and to understand the computational models. In response to the question, *JFLAP+backend should allow users to resubmit until correct*, 21 students responded with 20 students strongly agreeing and 1 student agreeing. Student responses to the question *How often did you resubmit until correct?* were *always* (3 students), *most of the time* (13), *about half the time* (3), and *rarely* (2). For the consenting students, there were 629 pairs of (homework problem, student), 528 of which converged to a correct answer for the problem, leading to an 84% persistence-until-correct rate.

Student free-text responses about resubmission until correct were positive, e.g., *sometimes* [JFLAP+server] *would inform me of something I overlooked and I was able to make the adjustments*. Students also wrote *Resubmitting until correct allowed me to truly grasp some of the concepts* and *It's easy to miss little details in the design of the automata, so it's extremely helpful to get direct, immediate feedback.*

ACKNOWLEDGMENTS

We thank the SIGCSE 2023 anonymous referees for helpful comments. We thank Ting Cao and her students for participating. We thank our advisory board: Douglas Baldwin, Joan Lucas, and Susan Rodger. Research supported in part by NSF grant DUE-1819546.

REFERENCES

[1] I. Bezáková, K. Fluet, E. Hemaspaandra, H. Miller, and D. E. Narváez. 2022. Effective Succinct Feedback for Intro CS Theory: A JFLAP Extension. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. 976–982.

[2] S. H. Rodger and T. W. Finley. 2006. *JFLAP: An Interactive Formal Languages and Automata Package*. Jones and Bartlett.

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576287