Prototype of an Automated Feedback Tool for Intro CS Theory

Ivona Bezáková
Rochester Institute of Technology

Edith Hemaspaandra
Rochester Institute of Technology

Aryeh Lieberman
Rochester Institute of Technology

Hannah Miller
Rochester Institute of Technology
Rochester, New York, U.S.A.

David E. Narváez
Rochester Institute of Technology
Rochester, New York, U.S.A.

ABSTRACT

Computing theory is an important part of computer science education, introducing students to computational models of increasing power to study possibilities and limitations of computation. The subject is, however, very abstract and mathematical, and students often struggle with it. Students must master various computational models, but there is often a lengthy delay from the time a model is introduced until a student gets feedback on their related assignment. During this time, the course has typically moved far ahead, and students become progressively more lost. To alleviate this problem, we developed a prototype of an automated feedback tool for CS theory, which extends the widely used JFLAP software. Our tool currently handles student submissions of deterministic and non-deterministic finite automata, regular expressions, context-free grammars, and push-down automata homework, where an instructor specifies the target language and the students receive immediate feedback on their submissions. Currently, for incorrect submissions, the feedback is in the form of a "witness" string, specifying a string on which the submission fails. Beyond regular languages, our tool attempts to solve undecidable problems; fortunately, the undecidability does not occur on typical homework assignments. We are collecting preliminary evaluation data from students using the prototype tool in their course. In our future work, we will analyze the data, and we aim to produce automated partial credit (along with the witness feedback) using SAT and QBF solvers.

CCS CONCEPTS

• **Social and professional topics → Student assessment**; • **Theory of computation →** *Grammars and context-free languages*;

KEYWORDS

formal languages and automata; models of computation

1 BACKGROUND AND RELATED WORK

Computing theory is difficult for beginner students since many of the concepts are complicated and abstract. In introductory theory courses, students are asked to construct finite automata, regular expressions, context-free grammars, push-down automata, and Turing machines that recognize certain languages. JFLAP (www.jflap.org) is a popular tool for students to interact with these concepts. We have developed an extension to JFLAP that accepts homework submissions from students, grades the solution, and provides a witness string when the submission is incorrect. An alternative tool, Automata Tutor (https://automata-tutor.model.in.tum.de/), provides similar functionality, but its scope is limited to regular languages [1, 2].

2 METHODS AND RESULTS

The tool that we have developed is an extension to JFLAP that connects with a grading server. This allows students to solve assignments using JFLAP and then use our extension to submit their solution for a particular problem to our grading server. Then, our grading server offers feedback of whether the solution is correct and a witness string if the solution is incorrect. For equivalence checking of context-free languages, which is an undecidable problem, we rely on the study of many homework assignments and submissions, where for each incorrect solution there was a witness string of length at most 10 [3].

In Fall 2019, we are running a preliminary evaluation of the efficacy of our feedback tool. Two synchronized sections of the Introduction to Computer Science Theory course share the same instructor, course delivery style, provided materials, homework assignments, analogous midterm exams, and a common final exam. The control section submits their homework solutions using traditional means while the study section is required to use our tool on selected homework assignments. We survey students on their experience in the course and on using our tool, collect their grade data, and monitor their use of our tool.

3 FUTURE WORK

In Spring 2020, we will analyze the evaluation data and use this to improve our tool. In Fall 2020, we will run a larger-scale study using the improved tool and investigate providing partial credit using SAT and QBF solvers. See https://cs.rit.edu/~afct/ for updates.

ACKNOWLEDGMENTS

We thank the anonymous referees and Kim Fluet for helpful comments. We thank Aaron Deever and his students for their participation. Research supported in part by NSF grant DUE-1819546.

REFERENCES

[1] Rajeev Alur, Loris D'Antoni, Sumit Gulwani, Dileep Kini, and Mahesh Viswanathan. 2013. Automated grading of DFA constructions. *IJCAI 2013*, 1976–1982.

[2] Loris D'Antoni, Dileep Kini, Rajeev Alur, Sumit Gulwani, Mahesh Viswanathan, and Björn Hartmann. 2015. How Can Automatic Feedback Help Students Construct Automata? *ACM Transactions on Computer-Human Interaction* 22 (2015), 9:1–9:24.

[3] Jessica Sorrell. 2015. CFG Solver. https://github.com/hatgirl/CFGSolver. (2015).

SIGCSE '20, March 11–14, 2020, Portland, OR, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6793-6/20/03.
https://doi.org/10.1145/3328778.3372598