Effective Succinct Feedback for Intro CS Theory: A JFLAP Extension

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

Computing theory is often perceived as challenging by students, and verifying the correctness of a student's automaton or grammar is time-consuming for instructors. Aiming to provide benefits to both students and instructors, we designed an automated feedback tool for assignments where students construct automata or grammars. Our tool, built as an extension to the widely popular JFLAP software, determines if a submission is correct, and for incorrect submissions it provides a "witness" string demonstrating the incorrectness.

We studied the usage and benefits of our tool in two terms, Fall 2019 and Spring 2021. Each term, students in one section of the Introduction to Computer Science Theory course were required to use our tool for sample homework questions targeting DFAs, NFAs, RegExs, CFGs, and PDAs. In Fall 2019, this was a regular section of the course. We also collected comparison data from another section that did not use our tool but had the same instructor and homework assignments. In Spring 2021, a smaller honors section provided the perspective from this demographic. Overall, students who used the tool reported that it helped them to not only solve the homework questions (and they performed better than the comparison group) but also to better understand the underlying theory concept. They were engaged with the tool: almost all persisted with their attempts until their submission was correct despite not being able to random walk to a solution. This indicates that witness feedback, a succinct explanation of incorrectness, is effective. Additionally, it assisted instructors with assignment grading.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Theory of computation** → **Formal languages and automata theory**; **Models of computation**.

KEYWORDS

Automated feedback, Introductory Computing Theory, Minimal intervention, JFLAP extension

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499416

ACM Reference Format:
Ivona Bezáková, Kimberly Fluet, Edith Hemaspaandra, Hannah Miller, and David 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 V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499416

1 INTRODUCTION

Computing theory analyzes various computational models and their computational power, in order to understand the amount of resources needed to solve real-world problems. Due to the abstract nature of the models and their analyses, these concepts are difficult for many beginner students. As an important step towards the understanding of these concepts, students in introductory computer science theory courses construct computational models such as deterministic finite automata, nondeterministic finite automata, regular expressions, context-free grammars, and pushdown automata (DFAs, NFAs, RegExs, CFGs, and PDAs). The most popular graphical interface for students to interact with the models [8] is the Java Formal Languages and Automata Package (JFLAP) [25–27].

We built an extension to JFLAP, which we named DAVID (Didactic And Visual Interface for Development), that accepts submissions of instances of DFAs, NFAs, RegExs, CFGs, or PDAs from students and immediately verifies the correctness of each submission. In particular, the DAVID extension consists of an extended JFLAP, see Figure 1, and a feedback server. The instructor sets up a problem on the server by choosing the computational model, describing in words or math notation the target language (the set of inputs that should be accepted by the student's submission), and providing a JFLAP instance of the model that accepts the language (a correct solution). Each student creates their own JFLAP instance of the model, aiming to accept the target language, and submits it through DAVID. The DAVID extension sends the submission to our feedback server, which automatically checks it against the instructor's correct solution; the server then provides immediate feedback to the student whether the submission is correct or not. If not, the server also provides an input string (a "witness") on which the submission differs from the instructor's solution. Essentially, the server provides a minimal reason why the submission is incorrect, prodding the student to reflect on what went wrong without oversteering them towards a specific way of solving the problem. These problems can be typically solved in many, very different, yet all correct ways. We note that the development of this system was a very substantial effort, one that will benefit other instructors as we will make DAVID publicly available.

We used the DAVID extension in the Introduction to Computer Science Theory course at a university in two academic terms, Fall 2019 and Spring 2021. Our study aimed to answer the following research questions:

RQ1 What is the effect of using the DAVID extension on students' perceptions of their learning and on their behavior when solving homework questions?

RQ2 What is the effect of using the DAVID extension on students' performance when solving the homework questions?

RQ3 How do instructors benefit from the DAVID extension?

Our investigation had multiple encouraging outcomes: The majority (about 55%) of students agreed that the use of the DAVID extension helped them to solve the DFA, NFA, and RegEx homework questions (and about 20% of students were neutral), while the percentage shifted to about 40%/40% for agreeing/neutral students for CFGs and PDAs. Moreover, for each of the computational models, around 40%-45% of students indicated that the extension helped them to understand the concept of that model. This is very promising: the students felt that the DAVID extension helped them to not only solve a specific homework question, but to understand the overall concept of the abstract computational model. The students who used the DAVID extension were engaged with it, consciously modifying their submissions until their answers were correct (we call this behavior "persistence"). These were not random modifications without thought—the students in the focus group elaborated on their process and explained that getting to a correct solution by randomly modifying their answers is close to impossible! We argue, therefore, that witness feedback is the most appropriate feedback for the types of assignments we target.

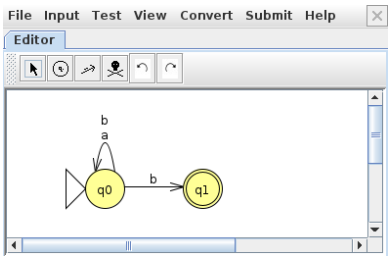

Figure 1: JFLAP with the DAVID extension "Submit" option.

2 LITERATURE REVIEW

Hattie and Timperley [19] define *feedback* as "information provided by an agent (e.g., teacher, peer, book, parent, experience) regarding aspects of one's performance or understanding." In this work, we study *intermediate feedback*, a type of *formative feedback* [28].

Automata Tutor (AT) [15] is an alternative feedback tool for computing theory, developed in parallel with our tool. In Fall 2019, version 2 of the AT performed equivalence checking of regular languages, including witness feedback, but not CFG and PDA equivalence checking [1, 12]. In mid-2020 a significantly expanded version 3 was released [11] that added context-free languages (and other features unrelated to our work). For model construction assignments, the AT's feedback includes more expansive hints to the student and automated grading, providing a numerical score for the current submission. In contrast, and by design, the DAVID extension feedback is succinct, avoiding over-guiding the students, and this type of feedback is the context for our educational research.

Another interesting feedback tool for CS theory is the very recent work of Mohammed, Shaffer, and Rodger [22], an e-textbook with simulations and auto-graded exercises. The aim of this excellent work is also different than ours: It is a full-fledged e-textbook, where the auto-graded exercises are a part of the work but not the main focus. The automated feedback is done via a set of test strings, analogous to unit tests in software development (for their use in introductory computing see, for example, [6, 29, 34]). In particular, the student's submission is checked against a (relatively small) set of input strings, and for each of these strings the system verifies whether it is accepted correctly by the student's submission. An interesting feature of the system (unlike in traditional programming unit testing) is that the set of test strings is randomly generated with each submission. However, it might be that a student's submission is missing a special case, which is unlikely to be randomly generated, so the unit-testing approach might proclaim an incorrect submission as correct. In contrast, our approach will find the special case input on which the submission fails.

Lastly, we consider work related to the DAVID extension implementation. Our feedback server needs to decide equivalence between the student's submission and the instructor's answer. For regular languages, DFA equivalence can be determined in near-linear time [18, 20], which Norton [23] implemented in JFLAP-compatible code, producing a witness string if the two DFAs are not equivalent. Since CFG equivalence is an undecidable problem, Sorrell used CFGAnalyzer [2] to experimentally show that checking all strings up to length $k = 10$ suffices for typical homework assignments. Sorrell's work included integrating the PicoSAT solver [3] into CFGAnalyzer; this work is the CFGSolver software [31] used by the DAVID extension. To be conservative, our extension checks all strings up to length $k = 15$.

3 RESEARCH DESIGN AND METHODOLOGY

Our research is a mixed-methodology [14, 21, 32] study of the DAVID extension to investigate students' perceptions, behavior, and performance when using the tool to solve introductory CS theory homework problems. In addition to this student-centered focus, we aim to understand the instructor benefit when the DAVID extension is part of CS theory homework. Our project team consists of researchers with doctorates in the fields of CS theory as well as STEM education research (with focus on STEM in higher education).

Course setting. In our research context, a mid-sized, technology-focused university, several sections of the Introduction to Computer Science Theory course are offered each term and are taught by different instructors, where each instructor assigns their own homework sets. The sections are capped at 40 students. In Fall 2019, two of those sections were selected for our research because they had the same instructor and were of similar size, around 37 students each.

The instructor agreed to require Section 1 to use the DAVID extension on 5 homework questions (of 55 total), that target the students' understanding of fundamental computational models, the grasp of which is necessary in order to move to more sophisticated course topics. Students in Section 2 had the option of solving the

questions on paper or using JFLAP, which is the standard practice for that instructor. Sections 1 and 2 shared the same course delivery style, homework assignments, had similar midterm exams, identical final exams, similar size and demographic profile. A third section in Spring 2021, with a different instructor, also required students to use DAVID when solving relevant homework questions. While Section 1 (DAVID req) and Section 2 (JFLAP opt) were typical offerings of the course, Section 3 (DAVID req-Honors) was a smaller honors section of 12 students, providing an opportunity to gain additional insights about students' use and thoughts of the tool. All three sections followed their usual course delivery style and therefore had minimal time dedicated to introducing the tools, DAVID for Sections 1 and 3, and JFLAP for Section 2. Both instructors are very experienced and have taught this course for many years.

Data collection. We collected data from students in all three sections. Data sources for Sections 1 and 2, in Fall 2019, included student homework submissions, homework grades, course grades and student survey responses. Both sections were surveyed twice, once after the first three targeted homework questions, and again after the last two targeted homework questions. Both sections provided demographic information and their perceptions about their self-perceived computer science and math abilities. Additionally, Section 1 (DAVID req) was surveyed about their perceptions of the DAVID extension, their experiences with the extension, their thoughts on automated feedback in general and whether they used any other tools. Section 2 (JFLAP opt) was surveyed about whether they used any tool to complete the homeworks and their perceptions of that tool, if one was used.

Both sections were graded using the same homework rubrics and by the same experienced grader. Since the extension provides feedback about the correctness of the submission (whether it corresponds to the desired language or not) but it does not assign a numerical grade, the questions targeted by the DAVID extension were graded manually. It is of note that when the instructor imposed additional requirements on the submission, for example, "your NFA should not be overly complicated," despite the DAVID extension judging a submission as correct, the grader could have subtracted credit for this unnecessary complexity ("insufficient nondeterminism").

In addition to the survey responses from students in Section 1 and Section 2, in Spring 2020 students in Section 3, the honors section, who used DAVID for 6 of their 39 homework questions and 3 additional practice problems, participated in an in-depth focus group (because of the smaller class size) to explore their experiences with the DAVID extension. This 50-minute focus group was conducted by our STEM education researcher and not by the course instructor. The focus group data provided rich descriptions of student experiences that are often not possible to collect with a survey instrument, however the focus group prompts were derived from the survey questions distributed to Section 1. The focus group format allowed for follow-up questions and deeper discussion on the topics, as well as the opportunity for the students to volunteer information not directly related to the prompts. Consequently, the focus group provided important student data about persistence, partial credit, and motivation to resubmit, using the DAVID extension.

We also monitored how the students in Sections 1 and 3 used the DAVID extension in terms of number of unique re-submissions and the trajectory of the changes to their re-submissions. Lastly, to determine benefit to the instructor, we interviewed the instructor for Section 1 and Section 2, and the instructor for Section 3 about their perspectives on the efficacy and impact of the DAVID extension. It should be noted that all participants in this study gave informed consent to have their data included in the research, and with the expectation of anonymity and confidentiality.

Data analysis. The quantitative data (Likert-type survey responses on a 5 point scale), student grades, and homework submissions) were analyzed using SPSS statistical software to report summary descriptive analyses as well as any statistically significant relationships among the data. The qualitative data (student open-ended survey comments, student focus group transcript and instructor interview transcripts) were analyzed using a modified grounded-theory approach [9, 16] to data analysis.

Homework questions. In Fall 2019 the DAVID extension was used for one homework question on each of the following computation models: DFA, NFA, RegEx, CFG, and PDA (to minimize the burden on the volunteer instructor). In Spring 2021 the homework questions also targeted these computational models and, along with additional practice questions, were included in the DAVID extension. For illustrative purposes we are including the statements of the DFA and CFG questions from Fall 2019:

(1) *DFA.* Draw the state diagram of a finite automaton that accepts the language of all strings over $\{a, b\}$ that contain at least 2 b's and do not contain the substring bb. In other words, a string is accepted only if both conditions hold. Your finite automaton should not be overly complicated.
(2) *CFG.* Give a CFG that generates the language of all strings over $\{0, 1\}$ that have more consecutive 0's at the beginning of the string than consecutive 1's at the end of the string. (For example, the following strings are all in the language: {0, 001, 00001010101010111, 0111111110}. The following strings are all not in the language: {ϵ, 01, 10, 0011, 0010000000111}.)

As seen from these examples, the level of difficulty of the homework questions in our research study was relatively high, since challenging questions provide an opportunity to provide feedback to the students on their understanding of the topic [5, 17, 33]. Such questions are tricky, not only for the students to solve, but also for an instructor to verify whether a given submission is correct. As such, we believe that more difficult questions are the best context in which to evaluate the usefulness of an automated feedback tool.

4 RESULTS

In total, 77 students voluntarily participated: 31 of 38 students in Section 1, 36 of 37 in Section 2 and 12 of 13 students in Section 3.

4.1 RQ1: Student perceptions and behavior

The students in Section 1 (DAVID req) were surveyed about their agreement with the following statements using a 5-point Likert scale ranging from Strongly disagree to Strongly agree:

S1 The DAVID extension helped me solve the homework question for {DFA, NFA, RegEx, CFG, PDA}.
S2 The DAVID extension helped me understand {DFAs, NFAs, RegExs, CFGs, PDAs}.

Table 1: Section 1 survey responses for S1 and S2 across all five computational models. (S)A: (Strongly) Agree, N: Neutral, (S)D: (Strongly) Disagree.

	S1: Solve homework			S2: Understand concept		
	(S)A	N	(S)D	(S)A	N	(S)D
DFA	54.9%	19.4%	25.7%	41.9%	29.0%	29.1%
NFA	51.6%	22.6%	25.8%	45.2%	25.8%	29.0%
RegEx	60.7%	21.4%	17.9%	46.4%	32.1%	21.5%
CFG	39.3%	42.9%	17.8%	42.8%	35.7%	21.5%
PDA	42.9%	39.3%	17.8%	39.3%	42.9%	17.8%

We report the survey response percentages in Table 1. For space reasons we combined the two positive responses into one group and we did the same for the two negative responses. The raw (anonymized) data and the detailed data analysis outputs for these as well as other responses are available upon request.

Overall, positive responses significantly outweigh the negative responses, indicating that the students found the DAVID extension helpful both for solving the homework questions as well as for understanding the computational models. For S1 there were at least twice as many students who agreed that DAVID helped them to solve the questions than those who disagreed, and this held for all computational models (for regular expressions this contrast was more than threefold). Following a similar trend, most students from the focus group agreed that the DAVID extension helped them solve the homework questions, e.g. "That's a really good way to facilitate the process of figuring out the problem step by step. It's confirming you already know but it gives you more practice with figuring out how to come to a complete solution from a partial solution. Most of the time, you won't necessarily have a complete solution at the very start, especially with a difficult problem. The DAVID extension helps you get to the final point and practice those important skills." (SP2021-FG, Student 07).

For S2, the ratio of agreeing and disagreeing students is about two for each of the models, and overall about 40% of students felt that the extension helped them to understand each of the concepts. We find this very encouraging, since many of these students might be those who need extra help with understanding the material. The percentage of students who disagreed with either S1 or S2 decreased as the complexity of the computational model increased, while the students who felt neutral increased. In response to S2, the majority of students from the focus group agreed that using the DAVID extension helped them understand the concepts better, e.g., "Being able to see you're wrong and if you spend time and work to fix that you better understand your problems and you can learn." (SP2021-FG, Student 02). When asked if future course offerings should use the DAVID extension, 67% of students in Section 1 agreed or strongly agreed that it should be used and some provided open-ended comments supporting this preference, "[The] DAVID extension is a very helpful tool. It helped me understand DFA[s] and NFA[s] better." (FA2019 Survey1, Student 21). Students from the Section 3 focus group also agreed with this sentiment.

In addition, we saw high engagement with the extension: on average, students submitted 9 times per homework question. Also,

Table 2: Persistence percentages for Section 1.

DFA	NFA	RegEx	CFG	PDA
92.0%	66.7%	76.9%	77.3%	88.0%

a high percentage of students, as shown in Table 2, *persisted* with their solution attempts until the extension reported "Correct."

We discussed the phenomenon of persistence we observed both as a research team and with our external advisory board. These discussions led us to consider the following scenarios for the usage of the DAVID extension: either allow students to resubmit as many times as they wish, or limit the number of submissions; either provide just correct/incorrect feedback with a witness string, or also include a numerical partial credit grade. Subsequently, on the second survey we also asked the Section 1 students about their opinions on resubmission, partial credit, and correct/incorrect feedback. They indicated, on a 5-point Likert scale, their agreement/disagreement with the following statements:

- **S3** The DAVID extension should allow users to resubmit until correct.
- **S4** Assignments submitted via the DAVID extension should be graded to allow partial credit.
- **S5** Assignments submitted via the DAVID extension should be graded as correct/incorrect (i.e., without partial credit).

Students overwhelmingly agreed (> 95%) that the DAVID extension should allow users to resubmit until correct, and they also agreed (71.4%) that assignments should be graded to allow partial credit, but they disagreed (60.6%) that assignments should be graded as strictly correct or incorrect. They felt substantially stronger about the option to resubmit than about partial credit. We also asked the focus group from Section 3 for their input about resubmission and partial credit. These students overwhelmingly supported resubmission but not partial credit, "But here I think [feedback] helps to understand where you need to study something more and develop more skills without necessarily being immediately penalized the first time you're wrong, or something…The feedback server is really important for that." (SP2021-FG, Student 04).

4.2 RQ2: Student performance

For RQ2, *What is the effect of using the feedback tool on students' performance when solving homework questions?*, we compared the differences between Section 1 (DAVID req) and Section 2 (JFLAP opt) homework grades, see Figure 2.

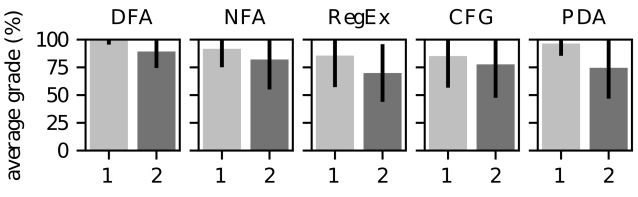

Figure 2: Homework grade average and standard deviation for the five targeted homework questions for Sects. 1 and 2.

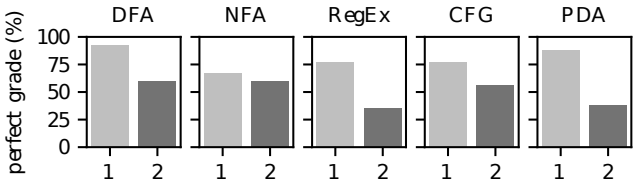

Figure 3: Percentage of students who earned a perfect grade on the five targeted questions for Sections 1 and 2.

Section 3, a small honors class, was not comparable to the larger Sections 1 and 2; subsequently, Section 3 grades were not included in the statistical analysis. Statistical analysis of Sections 1 and 2's homework grades was conducted using ANOVA with a threshold of $p < 0.050$. On three of the five targeted homework questions, Section 1's higher assignment score was statistically significant: on the DFA ($p = 0.001$), RegEx ($p = 0.030$), and PDA ($p < 0.001$) questions. For the other two targeted homework questions, NFA and CFG, there was no statistically significant difference between the sections. One could expect the section that received immediate, minimal feedback about the correctness of their submissions, and had the option to resubmit, to score higher. However, we have indications that Section 1 (DAVID req) , even though it was a large ~40 student section, was academically less proficient than Section 2 (JFLAP opt) , also a large ~40 student section. We find it noteworthy that the academically weaker section still scored higher, with statistical significance, for three of the models, and it is encouraging that there was no statistically significant difference between Section 1 and Section 2 scores on the NFA and CFG homework questions. We also note that Section 1 strongly outperformed Section 2 with respect to the percent of perfect homework grades for the targeted homework questions, see Figure 3.[1]

4.3 RQ3: Instructor benefit

Recall that the focus of our work is on developing a homework feedback server for students, and our work is not about automatic grading; we discuss this choice further, putting it in perspective with student perceptions from Section 4.1 as well as with instructor feedback, in Section 6. Nevertheless, we did collect data from instructors to determine their perceptions of instructor benefit. Both instructors agreed that getting the feedback of the DAVID extension on the student submissions allows the instructor to first check whether the DAVID extension deems the submission as correct (it accepts the correct language). In other words, the instructor has the option of automatically triaging the submissions into correct/incorrect language—we refer to this benefit as *grading triage*.

Grading of correct submissions is much faster and easier (even if one still needs to check for additional constraints such as "sufficient nondeterminism") than grading of incorrect submissions. Manual verification of a correct solution can be complicated for models such as regular expressions and CFGs; our instructors expressed appreciation that the DAVID extension checked the correctness of students' submissions for these questions. (The CFG question, which was especially challenging, had by far the most number of submissions to the DAVID extension.) And for incorrect submissions, the witness string feedback is a valuable resource to help an instructor see the error in the submission. Table 2 shows that an overwhelming percentage of submissions are correct and thus triaged into the "easy-to-grade" pile (see also Figure 3 for contrast when DAVID is not used). A student in the focus group even remarked that they thought the DAVID extension "[makes] it easier for the instructor to grade" (SP2021-FG, Student 02).

[1]The Section 1 NFA percentage is low compared to the other questions because the NFA submissions through the DAVID extension (at the time) were not checked for "sufficient" nondeterminism. This means that submissions that received feedback as "correct" (i.e., the submission was an NFA that accepted the correct language) did not necessarily earn a perfect grade when manually graded by an experienced instructor.

5 LIMITATIONS AND THREATS TO VALIDITY

After student registration in Fall 2019, we randomly chose one section to require the use of the DAVID extension (Section 1) and the other section to not use it (Section 2), but with the option to use JFLAP. The sections were scheduled back to back and at times generally favored by students (not very early or late in the day). However, we have reason to believe that, despite our best efforts, considerations beyond our control might have influenced the composition of these two sections. For example, if there is an advanced math course that coincides with one of the sections, all the interested students will navigate to the other section, potentially making Section 1 and Section 2 of unequal academic strength. As for Section 3 (Spring 2021), because it was a dedicated honors theory course with typically higher student performance, we omitted these grades from the statistical analysis.

The course instructors kindly agreed to let us conduct our research in their course sections with already packed course content. For Fall 2019, we planned to use the overall degree GPAs as well as the exam and course scores as proxies to measure the relative academic strength of the two sections. However, our IRB (Institutional Review Board) unexpectedly did not allow us to report the average student GPA in each section, leaving us to compare Section 1 and Section 2 only by their exam and course scores, the instructor's perceptions, and the students' self-reported perceptions.

The final exam was identical for both sections, and for the final exam grade, the Section 2 (JFLAP opt) mean was 79.8% (SD = 17.0%) for 35 students who consented to participate in the study, and Section 1 (DAVID req) mean was 70.3% (SD = 12.9%) for 29 students who consented to participate in the study. This difference was statistically significant ($p = 0.014$) and therefore is not likely due to chance. For the overall course grade, Section 2 performed better than Section 1 with respective means of 84.9% (SD = 9.9%) and 78.0% (SD = 11.0%), which was statistically significant ($p = 0.011$). With only about 10% of homework questions targeted by DAVID, we did not expect to see measurable impact on the students' exam or course scores. Therefore, while not ideal, we believe exam and course grades are good metrics to indicate the relative academic strength of the sections. Additionally, during the interview, the course instructor said, "I do think it's probably correct that just kind of as an average performance, Section 2 was a little sharper [than Section 1]."

Curiously, when Sections 1 and 2 students were surveyed for their self-perceptions in terms of their enjoyment of the course, their struggle in the course, and their struggle in CS and math

courses, there were no statistically significant differences between the two sections. Students responded with their agreement, or disagreement to these statements using the same Likert scale as before: 1, strongly disagree to 5, strongly agree. When asked to respond to the statement 'I enjoy the content of this course', the means for Section 1 and Section 2 were 4.18 and 4.00 respectively, both firmly in the 'agree' rating. For the statement 'I am struggling in this course' the means were 2.82 and 2.79, approaching neutral; for 'I typically struggle with CS courses' the means were 2.14 and 2.24, closer to disagree; for 'I typically struggle with mathematics in my courses' the means were 2.29 for both sections. As such, these self-reported perceptions did not help much in confirming that the sections were of different relative academic strength.

6 DISCUSSION AND CONCLUSIONS

We have seen high engagement (RQ1) of the students with our automated feedback tool, with positive perceptions (RQ1) and increased performance (RQ2), and we have also documented benefits to the instructor (RQ3). The goal of our tool is to provide the most useful feedback to the students to help them to learn the material. There are many types of feedback, from a simple correct/incorrect to step-by-step instruction on how to solve a problem. Another type of feedback, in contrast, is a partial credit grade—a single score that determines how far from correct the submission is. We believe that our choice of feedback, a witness string, is the most appropriate feedback to provide in our context.

The case for witness feedback. In educational literature, a minimal reasonable feedback, also called *minimal intervention*, has been found to promote better learning than more detailed feedback [35]. This type of feedback convincingly shows that the submission is wrong, but the feedback does not give any hints for fixing the submission. Creighton et al. say, "If feedback attempts to provide too much guidance, there is nothing left for the student to do or learn" [10]. Similar ideas are also in [13, 24].

Witness feedback satisfies these criteria, and is therefore the minimal reasonable feedback for our setting. This type of feedback mimics the feedback that an instructor or a tutor would give to a student seeking help by providing a short witness string showing where the student's attempt is incorrect, but not leading the student to the correct solution. We see this idea reflected in the statements from the Section 3 students during the focus group. When asked how they used the DAVID extension and the role the feedback had in their learning, they shared that they used the feedback from the server to help them regroup and rethink their approach to solving the problem, "Being able to see you're wrong and if you spend time and work to fix that, you better understand your problems and you can learn." (SP2021-FG, Student 10).

With witness feedback and unlimited resubmissions, we saw students' high engagement and persistence. One critique of allowing students to submit as many times as they like is that students may try to random walk to a correct solution. Because the witness does not give information about how to change the submission, the potential of randomly converging on the correct solution is highly unlikely. We asked the focus group about whether the DAVID extension, and the witness string feedback they received, could help them random walk to the solution. They overwhelmingly agreed that the feedback from the DAVID extension would not allow students to random walk to the solution, "I would be really shocked if I were to like use the resubmissions randomly guessing or fixing edge cases repeatedly to get a right answer and if it was right and not understanding – if I get a right answer it's because it made sense and I understood it." (SP2021-FG, Student 03) and "I think a good way to put it would be if you really didn't know what you were doing, there would not be a way to guess and get the right answer. You do have to have the foundation. But there's all sorts of different approaches." (SP2021-FG, Student 07). However, when giving more detailed feedback, encouraging random walks to a solution can be an issue. There is also a real risk of over-helping and leading the student to the solution step-by-step in such a way that the student contributes very little (even though the student may not realize this). Finally, more detailed feedback may encourage students to make local fixes that create more and more bloated submissions.

The case against partial credit. If students are chasing points, then they may be randomly trying to converge on the wrong thing (more points) rather than the right thing (the correct language). Cain and Babar [7] paraphrase Skinner [30], saying, "Attaching marks to an assessment task means that, from the student's perspective, the task will play a summative role and feedback is not seen as formative." They continue, "Interestingly, it has been reported that students pay more careful attention to feedback when there are no associated marks [4] or put another way 'marks' reduced student attention to formative feedback." We saw this idea in the Section 3 focus group when we asked students if they would prefer that the DAVID extension assigned a partial credit grade or score to their submission. The consensus was that adding a grade would lead students away from learning and towards being points-driven, "It [partial credit] just really opens it up to game the system." (SP2021-FG, Student 11). Witness feedback, as opposed to partial credit, might support competency/mastery-based grading and should be further investigated.

Conclusions. With our approach of minimal reasonable feedback, students have an unlimited number of retries with immediate witness feedback, and can also seek help from the instructor or tutors. We did ask the focus group how often they resubmitted and why. Their answer was encouraging. They, as a group, asserted that they resubmitted until they received feedback from the server that their submission was correct. When asked what drove them to resubmit, they shared that it was their not knowing how close they were to the correct answer as well as the motivation to be correct that fueled their persistence, "The feedback tells you if you're right or wrong. It's pretty binary. The motivation exists to be right." (SP2021-FG, Student 01). We saw evidence of this same persistence in Section 1 with the number of students who earned perfect scores on the homework questions, see Figure 3. And, we know from the focus group reflections that their correctness is likely not due to random walks to the solution, but to a scenario in which they must work out the solution with the minimal feedback they received and their own understandings of the concepts.

ACKNOWLEDGMENTS

We thank the SIGCSE reviewers and our advisory board: Doug Baldwin, Joan Lucas, and Susan Rodger for helpful comments and

suggestions. We thank Aaron Deever and the students for their participation. Research supported in part by NSF grant DUE-1819546. David E. Narváez is supported by NSF grant CCF-2030859 to the Computing Research Association for the CIFellows Project.

REFERENCES

[1] Rajeev Alur, Loris D'Antoni, Sumit Gulwani, Dileep Kini, and Mahesh Viswanathan. 2013. Automated Grading of DFA Constructions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* (Beijing, China) *(IJCAI '13)*. AAAI Press, 1976–1982.

[2] Roland Axelsson, Keijo Heljanko, and Martin Lange. 2008. Analyzing Context-Free Grammars Using an Incremental SAT Solver. In *Automata, Languages, and Programming*, Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 410–422. https://link.springer.com/chapter/10.1007%2F978-3-540-70583-3_34

[3] Armin Biere. 2008. PicoSAT Essentials. *J. Satisf. Boolean Model. Comput.* 4 (2008), 75–97.

[4] Paul Black and Dylan Wiliam. 1998. Assessment and Classroom Learning. *Assessment in Education* 5, 1 (March 1998), 7–74.

[5] John Bransford, Rodney R Cocking, and Ann L Brown. 2000. *How people learn: Brain, mind, experience, and school (expanded edition)*. National Academies Press. https://doi.org/10.17226/9853

[6] Kevin Buffardi, Pedro Valdivia, and Destiny Rogers. 2019. Measuring Unit Test Accuracy. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 578–584. https://doi.org/10.1145/3287324.3287351

[7] Andrew Cain and Muhammad Ali Babar. 2016. Reflections on Applying Constructive Alignment with Formative Feedback for Teaching Introductory Programming and Software Architecture. In *Proceedings of the 38th International Conference on Software Engineering Companion* (Austin, Texas) *(ICSE '16)*. Association for Computing Machinery, New York, NY, USA, 336–345. https://doi.org/10.1145/2889160.2889185

[8] Pinaki Chakraborty, P. C. Saxena, and C. P. Katti. 2011. Fifty Years of Automata Simulation: A Review. *ACM Inroads* 2, 4 (Dec. 2011), 59–70. https://doi.org/10.1145/2038876.2038893

[9] Mark A Constas. 1992. Qualitative analysis as a public event: The documentation of category development procedures. *American Educational Research Journal* 29, 2 (1992), 253–266.

[10] Susan Janssen Creighton, Cheryl Rose Tobey, Eric E. Karnowski, and Emily Roche Fagan. 2015. Providing and using formative feedback. In *Bringing math students into the formative assessment equation*. Corwin, Chapter 4, 109–152.

[11] Loris D'Antoni, Martin Helfrich, Jan Kretinsky, Emanuel Ramneantu, and Maximilian Weininger. 2020. Automata Tutor v3. In *Computer Aided Verification (CAV '20)*, Shuvendu K. Lahiri and Chao Wang (Eds.). Springer International Publishing, 3–14.

[12] Loris D'Antoni, Dileep Kini, Rajeev Alur, Sumit Gulwani, Mahesh Viswanathan, and Björn Hartmann. 2015. How Can Automatic Feedback Help Students Construct Automata? *ACM Trans. Comput.-Hum. Interact.* 22, 2, Article 9 (March 2015), 24 pages. https://doi.org/10.1145/2723163

[13] Jeanne D. Day and Luis A. Cordón. 1993. Static and dynamic measures of ability: An experimental comparison. *Journal of Educational Psychology* 85, 1 (Mar 1993), 75–82.

[14] Terry E Dielman. 2005. *Applied regression analysis: A second course in business and economic statistics*. Brooks/Cole Thomson Learning Belmont, CA.

[15] Loris D'Antoni, Martin Helfrich, Jan Kretinsky, Emanuel Ramneantu, and Maximilian Weininger. [n.d.]. Automata Tutor. https://automata-tutor.model.in.tum.de/

[16] Barney G Glaser and Judith Holton. 2004. Remodeling grounded theory. In *Forum qualitative sozialforschung/forum: qualitative social research*, Vol. 5.

[17] Carina Granberg. 2016. Discovering and addressing errors during mathematics problem-solving—A productive struggle? *The Journal of Mathematical Behavior* 42 (2016), 33–48. https://doi.org/10.1016/j.jmathb.2016.02.002

[18] David Gries. 1972. *Describing an algorithm by Hopcroft*. Technical Report Technical Report TR 72-151. Cornell University.

[19] John Hattie and Helen Timperley. 2007. The Power of Feedback. *Review of Educational Research* 77, 1 (2007), 81–112. https://doi.org/10.3102/003465430298487 arXiv:https://doi.org/10.3102/003465430298487

[20] John Hopcroft. 1971. *An $n \log n$ algorithm for minimizing states in a finite automaton*. Technical Report Technical Report STAN-CS-71-190. Stanford University.

[21] R Burke Johnson and Anthony J Onwuegbuzie. 2004. Mixed methods research: A research paradigm whose time has come. *Educational researcher* 33, 7 (2004), 14–26.

[22] Mostafa Mohammed, Clifford A. Shaffer, and Susan H. Rodger. 2021. Teaching Formal Languages with Visualizations and Auto-Graded Exercises. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. ACM, 569–575. https://doi.org/10.1145/3408877.3432398

[23] Daphne A. Norton. 2009. *Algorithms for Testing Equivalence of Finite Automata, with a Grading Tool for JFLAP*. Master's thesis. Rochester Institute of Technology. https://scholarworks.rit.edu/theses/6939/

[24] Katrin Rakoczy, Petra Pinger, Jan Hochweber, Eckhard Klieme, Birgit Schütze, and Michael Besser. 2019. Formative assessment in mathematics: Mediated by feedback's perceived usefulness and students' self-efficacy. *Learning and Instruction* 60 (2019), 154–165. https://doi.org/10.1016/j.learninstruc.2018.01.004

[25] Susan H. Rodger. 1996. JFLAP: The Java Formal Languages and Automata Package. http://www.jflap.org/

[26] Susan H. Rodger and Thomas W. Finley. 2006. *JFLAP: An Interactive Formal Languages and Automata Package*. Jones and Bartlett Publishers, Inc., USA. http://www.jflap.org/jflapbook/jflapbook2006.pdf

[27] Susan H. Rodger, Eric Wiebe, Kyung Min Lee, Chris Morgan, Kareem Omar, and Jonathan Su. 2009. Increasing Engagement in Automata Theory with JFLAP. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (Chattanooga, TN, USA) *(SIGCSE '09)*. Association for Computing Machinery, New York, NY, USA, 403–407. https://doi.org/10.1145/1508865.1509011

[28] D. Royce Sadler. 1989. Formative assessment and the design of instructional systems. *Instructional Science* 18 (1989), 119–144. Issue 2. https://doi.org/10.1007/BF00117714

[29] Lilian Passos Scatalon, Jeffrey C. Carver, Rogério Eduardo Garcia, and Ellen Francine Barbosa. 2019. Software Testing in Introductory Programming Courses: A Systematic Mapping Study. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 421–427. https://doi.org/10.1145/3287324.3287384

[30] Kate Skinner. 2014. Bridging gaps and jumping through hoops: First-year History students' expectations and perceptions of assessment and feedback in a research-intensive UK university. *Arts and Humanities in Higher Education* 13 (09 2014), 359–376. https://doi.org/10.1177/1474022214531502

[31] Jessica Sorrell. 2015. CFGSolver. https://github.com/hatgirl/CFGSolver

[32] Anselm Strauss and Juliet Corbin. 1990. *Basics of qualitative research*. Sage publications.

[33] Lev Semenovich Vygotsky. 1978. *Mind in society: The development of higher psychological processes*. Harvard university press.

[34] Dee A. B. Weikle, Michael O. Lam, and Michael S. Kirkpatrick. 2019. Automating Systems Course Unit and Integration Testing: Experience Report. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 565–570. https://doi.org/10.1145/3287324.3287502

[35] Dylan Wiliam. 1999. Formative Assessment in Mathematics Part 2: Feedback. *Equals: Mathematics and Special Educational Needs* 5, 3 (Jan 1999), 8–11.