Seth Poulsen
sethpoulsen.github.io
sethp3@illinois.edu |*(559) 999-1482*

Research Statement

My research focuses on creating and evaluating software tools to support computer science education. Well designed software tools have the ability to provide huge learning gains for students [2, 18, 19, 20]. They also have the potential to increase equity and access to computing education by providing feedback to students who would not otherwise have access to it.

Proof Blocks, the subject of my dissertation, is now integrated into multiple online learning systems (PrairieLearn and Runestone Interactive) and has been used by thousands of students and continues to be used by more every semester. I also have experience collaborating across departments and across Universities on large and impactful educational data analysis projects, and advising students on a variety of projects.

Proof Blocks

My thesis research focuses on Proof Blocks (proofblocks.org), a tool which enables students to construct mathematical proofs by dragging and dropping prewritten proof lines into the correct order [2, 3, 6, 7, 8, 9]. Proof Blocks is not meant to replace proof writing, but to scaffold students as they make the transition from reading proofs to writing their own. This work has been published in premier venues in computer science education including AIED, ITiCSE, and ICER (with an honorable mention for best paper award).

Proof Blocks Implementation To write a Proof Blocks question, an instructor specifies the lines of the proof and the logical dependence between lines of the proof, forming a directed acyclic graph (DAG). Any topological sort of the DAG is accepted as a correct answer to the problem. Because of the large space of possible solutions for these problems, it is computationally expensive to calculate the difference between an incorrect student solution and some correct solution, making it difficult to assign partial credit. I created and implemented a novel algorithm for finding the edit distance from an arbitrary student submission to some correct solution and benchmarked it on thousands of student submissions from multiple courses, showing its effectiveness. This grading algorithm is now used by thousands of students per semester, not just for Proof Blocks problems, but also for other types of drag-and-drop problems like Parsons Problems, task scheduling problems, and other problems where the dependence between items can be modeled as a DAG. Work on Proof Blocks implementation has been presented both at ITiCSE and at AIED, with more work in preparation [6, 7, 8].

Proof Blocks as Exam Questions Using data from the discrete mathematics class at the University of Illinois, I provided statistical evidence that Proof Blocks are easier than written proofs for students to solve and that Proof Blocks problems provide the same amount of statistical information about student knowledge as written proofs. To make this analysis as reliable as possible, we used data from quizzes that students took each week over the course of the entire semester, across many topics. We used item response theory, a branch of educational statistics that allows estimation of characteristics of both questions and students and is robust against missing data.

To gain greater insight into students feelings about Proof Blocks problems, we also provided students an anonymous survey to complete. Students felt that the Proof Blocks user interface was easy to use, and that the questions accurately represent their ability to write proofs. This work received an honorable mention for the best paper award at ICER [3], considered the premier venue for computer science education research.

Proof Blocks as a Learning Tool My research has shown that Proof Blocks is successful as a learning tool, helping students learn proof by induction four times faster than when doing an equivalent proof writing activity. By recruiting students from the discrete mathematics course and inviting them to complete extra learning activities, we were able to randomly assign students to different learning conditions in a way that was ethically sound. While randomized controlled trials are the gold standard for establishing causation in educational studies, they are rarely performed due to the difficulty of assigning students to different experimental conditions. The students participating in our study ($n = 332$) completed a pretest on proof writing and a brief (less than 1 hour) learning activity and then returned one week later to complete the posttest. Depending on the experimental condition that each student was assigned to, they either completed only Proof Blocks problems, completed some Proof Blocks problems and some written proofs, or completed only written proofs for their learning activity. We found that students were able to learn just as much by using Proof Blocks as by writing proofs from scratch, but in far less time. This work is under review at the SIGCSE technical symposium [2].

Educational Data Analysis

My depth and breadth of educational data analysis skills has led to many fruitful collaborations both within my department and beyond. With colleagues in my department, we took on the challenge of replicating a well known paper in computing education research. To do so, I analyzed data from hundreds of students in a CS1 class using structural equation modeling, and uncovered major methodological flaws in the original study, which posited a temporal relationship between learning certain types of programming tasks without having proper evidence [4]. I also analyzed years worth of grade data from our department to understand the impact that course scheduling has on student performance, using a multi-level model in order to properly model variance between courses and between sections of the same course [16]. With a few faculty in my department, I analyzed course data to estimate the effect of students working synchronously versus asynchronously on group work [5], and the types of errors that students make while learning to write database queries in various query languages [12, 13, 17]. Working with faculty across the computer science, mechanical engineering, and mathematics deparment, I helped to measure the impact of a project to redesign the linear algebra course to incorporate computational labs [1]. I worked with faculty from over 20 different universities to collect and analyze data to validate the cybersecurity concept inventory, a measure of cybersecurity knowledge [10, 15]. Each of these projects required knowledge of different aspects of educational statistics and data analysis, and working together with my collaborators to understand the context the data came from. My depth and breadth of educational statistics will continue to be an asset for me as I continue to foster new research collaborations.

Dissemination of My Work

After integrating Proof Blocks into PrairieLearn, an open source homework and exam platform, I became the maintainer of the portion of the codebase that handles all drag-and-drop questions including Proof Blocks, Parson's problems, and other types of drag-and-drop questions including task scheduling problems, problems for helping students understand the order of operations in networking protocols, and other types of questions written by creative instructors. Maintaining this open source codebase has been a benefit to my research in multiple ways. It has made it easier for people to adopt my work, so that now thousands of students each semester across more than a dozen universities are benefiting from it. It has also helped me expand my network of research collaborators—I have been able to receive data for my research from instructors across five different universities. My position in the open source community also enables me to mentor student contributions to open-source software—a great way to help students grow in their software development skills while making positive contributions to the world. I am also working on integrating question sharing features into PrairieLearn to make it even easier for my work to spread.

This summer I integrated my novel grading algorithms for Proof Blocks into Runestone Interactive, an open source platform for building interactive textbooks used at thousands of institutions. I am in the process of working with collaborators to get Proof Blocks problems integrated into an open online textbook for discrete mathematics, giving anyone the opportunity to use Proof Blocks as a learning tool while working through a free open source textbook, even those without the opportunity to access education through a university.

Future Work

Tools for CS Theory Education While my thesis research has demonstrated the utility of Proof Blocks, there are many promising variations of Proof Blocks to try, and many more research questions to answer. For example, does including distractors in Proof Blocks problems improve learning gains? What about leaving blanks inside the blocks for students to fill in, similar to faded Parson's problems [14]? How can we make Proof Blocks work best for people who are blind or visually impaired? Because these are relatively small changes to make to existing open-source software, they would make great research projects for students relatively new to software engineering. I would advise advanced students in taking on bigger projects such as designing entirely new interactive learning tools, as there are more areas of computing theory education where students are not able to receive scaffolding and fast, automated feedback on their work. We will evaluate these tools in user studies, using similar methods to what I have done in my PhD work [2, 3]. Students can also work on analyzing data from student submissions and design additional learning studies to understand if the tools we build are working well for student learning and assessment.

Meta-analyses on CS1 Interventions for Broadening Participation in Computing Over the last few decades, various interventions have been studied for improving retention in CS1 course, especially of students from underrepresented groups, but there are not yet many meta-analyses to attempt to aggregate the results of these studies. Performing meta-analyses is a way to increase confidence in research results, estimate the true effect of an intervention over many studies and contexts, and estimate the impact of publication bias on the literature [21]. Very few meta-analyses exist in the computing education research literature, though conducting these meta-analyses is critical to ensuring confidence in the results of CS education research. My strong background in educational statistics give me the insight I need to perform these meta-analyses.

Some example topics in computer science education that are candidates for meta-analysis are the use of peer instruction in computer science education and the use of spatial skills interventions to improve students programming ability [11]. These interventions show promising results toward increasing the presence of women and underrepresented minorities in computer science, but without doing a meta-analysis to estimate the average impact of such interventions across many studies, it is hard to say for certain what their impact is. Performing meta-analyses can help researchers and practitioners gain a better understanding of the costs and benefits of such interventions, so that they can be implemented in the most effective way possible to improve representation in computing. These will be great projects for students to join because there is a low barrier to entry—as early as their freshmen or sophomore years, students could contribute by reading the research literature to extract the statistics needed for meta-analysis. More experienced students could contribute to the data analysis, and identify topics for future meta-analyses.

References[1]

[1] Aysa Fan, **Seth Poulsen**, Mariana Silva, and Matthew West. Integrating computational tools into a first course in linear algebra. in preparation.

[2] **Seth Poulsen**, Yael Gertner, Benjamin Cosman, Matthew West, and Geoffrey L. Herman. Learning gains of proof blocks versus written proofs. in submission.

[3] **Seth Poulsen**, Mahesh Viswanathan, Geoffrey L Herman, and Matthew West. Evaluating proof blocks problems as exam questions. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, pages 157–168, 2021 **Best Paper Runner-up**.

[4] Max Fowler, David H Smith IV, Mohammed Hassan, **Seth Poulsen**, Matthew West, and Craig Zilles. Reevaluating the relationship between explaining, tracing, and writing skills in cs1 in a replication study. *Computer Science Education*, pages 1–29, 2022.

[5] Geoffrey L. Herman, Yucheng Jiang, Yueqi Jiang, **Seth Poulsen**, Mariana Silva, and Matthew West. An analytic comparison of student-scheduled and instructor-scheduled collaborative learning in online contexts. In *Proceedings of the 2022 American Society for Engineering Education Conference*. American Society for Engineering Education, 2022.

[6] **Seth Poulsen**, Shubhang Kulkarni, Geoffrey Herman, and Matthew West. Benchmarking partial credit grading algorithms for proof blocks problems. In *International Conference on Artificial Intelligence in Education*, pages 199–203. Springer, 2022.

[7] **Seth Poulsen**, Shubhang Kulkarni, Geoffrey Herman, and Matthew West. Efficient partial credit grading of proof blocks problems. *arXiv preprint arXiv:2204.04196*, 2022.

[8] **Seth Poulsen**, Mahesh Viswanathan, Geoffrey L. Herman, and Matthew West. Proof blocks: Autogradable scaffolding activities for learning to write proofs. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, ITiCSE '22, page 428–434, New York, NY, USA, 2022. Association for Computing Machinery.

[9] **Seth Poulsen**, Matthew West, and Talia Ringer. Autogenerating natural language proofs for proof education. In *The Coq Workshop*, 2022.

[10] **Seth Poulsen**, Geoffrey L Herman, Peter AH Peterson, Enis Golaszewski, Akshita Gorti, Linda Oliva, Travis Scheponik, and Alan T Sherman. Psychometric evaluation of the cybersecurity concept inventory. *ACM Trans. Comput. Educ.*, 22(1), oct 2021.

[11] Jack Parkinson, Ryan Bockmon, Quintin Cutts, Michael Liut, Andrew Petersen, and Sheryl Sorby. Practice report: six studies of spatial skills training in introductory computer science. *ACM Inroads*, 12(4):18–29, 2021.

[1]My name in bold, students I mentored underlined.

[12] Mei Chen, **Seth Poulsen**, Ridha Alkhabaz, and Abdussalam Alawini. A quantitative analysis of student solutions to graph database problems. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 283–289, 2021.

[13] Ridha Alkhabaz, **Seth Poulsen**, Mei Chen, and Abdussalam Alawini. Insights from student solutions to mongodb homework problems. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 276–282, 2021.

[14] Nathaniel Weinman, Armando Fox, and Marti A Hearst. Improving instruction of programming patterns with faded parsons problems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2021.

[15] Alan T Sherman, Geoffrey L Herman, Linda Oliva, Peter AH Peterson, Enis Golaszewski, **Seth Poulsen**, Travis Scheponik, and Akshita Gorti. Experiences and lessons learned creating and validating concept inventories for cybersecurity. In *National Cyber Summit*, pages 3–34. Springer, 2020.

[16] **Seth Poulsen**, Carolyn J. Anderson, and Matthew West. The relationship between course scheduling and student performance. In *Proceedings of the 4th Workshop on Educational Data Mining in Computer Science Education*, 2020.

[17] **Seth Poulsen**, Liia Butler, Abdussalam Alawini, and Geoffrey L Herman. Insights from student solutions to sql homework problems. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, pages 404–410, 2020.

[18] Barbara J Ericson, James D Foley, and Jochen Rick. Evaluating the efficiency and effectiveness of adaptive parsons problems. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*, pages 60–68, 2018.

[19] Barbara J Ericson, Lauren E Margulieux, and Jochen Rick. Solving parsons problems versus fixing and writing code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, pages 20–29, 2017.

[20] Greg L. Nelson, Benjamin Xie, and Amy J. Ko. Comprehension first: Evaluating a novel pedagogy and tutoring system for program tracing in cs1. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ICER '17, page 2–11, New York, NY, USA, 2017. Association for Computing Machinery.

[21] Harris Cooper. *Research synthesis and meta-analysis: A step-by-step approach*, volume 2. Sage publications, 2015.