Snowflake: Supporting Programming and Proofs

Oluwatobi Alabi
Grinnell College
Grinnell, IA, USA
alabijem@grinnell.edu

Anh Vu
Grinnell College
Grinnell, IA, USA
vuanh@grinnell.edu

Peter-Michael Osera
Grinnell College
Grinnell, IA, USA
osera@cs.grinnell.edu

ABSTRACT

Rigorous, mathematical reasoning, *i.e.*, proof, is the foundation of any undergraduate computer science education. However, students find mathematical proof exceedingly challenging, but also at the same time do not see its relevance to programming. We address these concerns with Snowflake, an educational proof assistant designed to help undergraduates overcome these difficulties when authoring mathematical proof. Snowflake does this by operating in a context where mathematical proof is introduced alongside programming in either a CS1 or CS2 context. The lens that we use to unite the two concepts is *program correctness*, a topic that immediately makes relevant the concept of formal reasoning as students are perpetually faced with the issue of whether their code is correct.

Snowflake is a proof assistant designed for the needs of undergraduates in courses that closely time programming and proof. It is a web-based application that helps students author proofs not only in the context of program correctness in-the-small, but also other topics found in discrete mathematics courses. We report on the design of Snowflake, the kinds of reasoning it enables, and our plans to deploy Snowflake in the classroom.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Theory of computation** → *Proof theory*; *Automated reasoning*.

KEYWORDS

Computer science education, Discrete mathematics, Proof assistant

ACM Reference Format:
Oluwatobi Alabi, Anh Vu, and Peter-Michael Osera. 2023. Snowflake: Supporting Programming and Proofs. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3545947.3576342

1 BACKGROUND AND DESIGN

Mathematical skills are essential in writing well-structured programs and designing efficient algorithms. However, proof-based mathematics courses are generally perceived by students as challenging and pointless [3]. The problem stems from not only the difficulty of the field but also our pedagogical approaches. While introductory CS courses occur in feedback-rich environments, theoretical mathematics courses do not. Instructors first explain proof concepts expecting students to absorb them fully. Then students follow the model on paper, only receiving limited, external feedback much later when the work is finally graded [3]. This methodology seems inadequate for students to fully absorb and become fluent in reading and writing mathematical proofs.

Snowflake addresses these concerns by supporting undergraduates authoring mathematical proof in the context of *program correctness* and *discrete mathematics*, situations that, when combined with appropriate pedagogy, give meaning to formal proof. Our work expands on prior work [1, 4] by directly supporting pedagogy in two courses, a CS1 and discrete mathematics course featuring program correctness to unite the worlds of programming and proof. Snowflake is built in Typescript as a browser-based application to allow for easier integration with a web interface and creation of a more accessible and interactive experience for students.

Snowflake features a minimal proof engine core for deductive reasoning that we instantiate to support particular proof domains. This core represents proofs in a tree-like structure built upon first-order logic and natural deduction. Students' proofs are translated into this minimal core, and Snowflake checks their validity using its internal logic coupled with an off-the-shelf SMT solver. Finally, Snowflake informs students about the correctness of their proof, and where they made mistakes.

In the context of program correctness, Snowflake allows students to reason about the behavior of programs written in a pure, functional subset of the Python programming language. Students use Snowflake to reason about the step-by-step execution of their program in a simplified style of operational semantics commonly found in the field of programming language theory [2].

2 CONCLUSION AND FUTURE WORK

We plan to utilize Snowflake in both a discrete mathematics and CS1 classroom in an upcoming semester to gauge its effectiveness in helping students understand, author, and validate well-formed and appropriate proofs. Additionally, we plan to continue improving Snowflake 's user interface to minimize user friction.

REFERENCES

[1] Graham Leach-Krouse. 2018. Carnap: An Open Framework for Formal Reasoning in the Browser. *Electronic Proceedings in Theoretical Computer Science* 267 (March 2018), 70–88. https://doi.org/10.4204/EPTCS.267.5 arXiv: 1803.03092.
[2] Benjamin C. Pierce. 2002. *Types and Programming Languages.* MIT Press.
[3] Nikki Sigurdson and Andrew Petersen. 2019. A Survey-based Exploration of Computer Science Student Perspectives on Mathematics. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education.* ACM, Minneapolis MN USA, 1032–1038. https://doi.org/10.1145/3287324.3287416
[4] David G. Wonnacott and Peter-Michael Osera. 2019. A Bridge Anchored on Both Sides: Formal Deduction in Introductory CS, and Code Proofs in Discrete Math. *arXiv:1907.04134 [cs]* (July 2019). http://arxiv.org/abs/1907.04134 arXiv: 1907.04134.

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

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576342