Using Knot Mosaics to Introduce Undergraduates to SAT

Hannah Miller
Rochester Institute of Technology

ABSTRACT

Knot mosaics are combinatorial representations of mathematical knots. Boolean satisfiability (SAT) is an important NP-complete problem, and SAT solvers are practical software implementations to solve SAT problems. SAT solvers are rarely used in undergraduate classes, and the learning curve for SAT is steep. For our contribution, we developed a lecture and a homework assignment for undergraduate students to work with SAT formulas, to draw knot mosaics, and to run SAT solvers for counting knot mosaics. The assignment has been developed over three semesters. The assignment includes a skeleton Python encoding with over 750 lines of Python code and function documentation. We surveyed the students about their experiences with and perceptions of the assignment as well as their comments for improving the assignment. On the survey, students overwhelmingly agreed that they learned about SAT and knots, and the students earned high grades on the assignment, showing that their perception of learning was true. Our future work includes clarifying one of the assignment problems where students struggled.

1 BACKGROUND & RELATED WORK

A mathematical *knot* is an embedding of a circle in 3D space, and a *link* is two or more knots. *Knot mosaics* [2] are arrangements of five unique tiles (Figure 1) and their rotations such that the curves make closed loops (Figure 2). SAT is a classic NP-complete problem. A *SAT formula* is a Boolean formula φ with Boolean variables x_i; logical operations AND ($\wedge$), OR ($\vee$), and NOT ($\neg$); and parentheses. A SAT solver finds a single *satisfying assignment*, which is an assignment of truth values such that φ is true. SAT solvers can be used to count knot mosaics [3].

2 METHODS

Building on [3], we designed a *SAT+Knots* assignment with four problems: (1) solve a SAT formula by hand, (2) draw knot mosaics by hand, (3) convert between knot mosaic tuples and SAT variables by hand, (4) use provided Python code[1] to count all small knots with several different SAT solvers (using PySAT bindings [1]). We gave the assignment to a class in Fall 2022. Out of 25 students, 14 students consented for their data to be used in our study. Our research questions are **RQ1** Did students' grades show that they learned about SAT solvers and knot mosaics? **RQ2** Did students perceive that they learned about SAT solvers and knot mosaics in general? **RQ3** What were students' perceptions of this specific SAT+Knots assignment?

[1] https://github.com/HM0880/knot-mosaics

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

ACM ISBN 978-1-4503-9433-8/23/03.
https://doi.org/10.1145/3545947.3576362

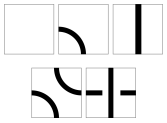

Figure 1: Unique tiles

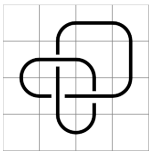

Figure 2: Trefoil knot

3 RESULTS & FUTURE WORK

To answer RQ1, most students earned full points for every problem, which is a proxy indicator that the students did learn. To answer RQ2, students agreed that they learned about SAT, mathematical knots, and knot mosaics (Table 1). To answer RQ3, free-text student comments were mostly positive, e.g., *I enjoyed problem 3. Being able to understand the transform by doing it by hand was nice because it let me understand what the program was doing for me.* However, students struggled with understanding the deeper concepts of solving a SAT formula. To address this issue, our future work includes guiding students with a concrete example.

Table 1: Student responses to *From the SAT+Knots lecture and the SAT+Knots assignment, I learned about [Category].* In every category, there was 1 response for *neutral* and 0 responses for *strongly disagree, disagree,* or *somewhat disagree.*

Category	Somewhat Agree	Agree	Strongly Agree
Theory of SAT	7	6	0
Applications of SAT	6	4	3
Encoding a SAT formula	7	4	2
Using SAT solvers	5	7	1
Mathematical knots	3	5	5
Knot mosaics	3	5	5
SAT+Knots	6	6	1

ACKNOWLEDGMENTS

Thanks to T.J. Borrelli for his feedback and to his students for responding to the survey questions. Thanks to A. Brilliantova for reading the draft. Partially supported by NSF grant DUE-1819546.

REFERENCES

[1] A. Ignatiev, A. Morgado, and J. Marques-Silva. 2018. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*. Springer, 428–437. https://doi.org/10.1007/978-3-319-94144-8_26
[2] S. J. Lomonaco and L. H. Kauffman. 2008. Quantum knots and mosaics. *Quantum Information Processing* 7, 2 (2008), 85–115.
[3] H. Miller. 2022. Enumerating Nontrivial Knot Mosaics with SAT (Student Abstract). In *AAAI*, Vol. 36. 13017–13018. https://doi.org/10.1609/aaai.v36i11.21645