

Enumerating Nontrivial Knot Mosaics with SAT (Student Abstract)

Hannah Miller

Golisano College of Computing and Information Sciences
Rochester Institute of Technology, Rochester, NY 14623
hm@mail.rit.edu

Abstract

Mathematical knots are interesting topological objects. Using simple arcs, lines, and crossings drawn on eleven possible tiles, knot mosaics are a representation of knots on a mosaic board. Our contribution is using SAT solvers as a tool for enumerating nontrivial knot mosaics. By encoding constraints for local knot mosaic properties, we computationally reduce the search space by factors of up to 6600. Our future research directions include encoding constraints for global properties and using parallel SAT techniques to attack larger boards.

Introduction and Related Work

A mathematical *knot* is an embedding of a circle in 3D space, and a *link* is two or more knots. Knots have applications in low-energy quantum physics (Hall et al. 2016). Lomonaco and Kauffman (2008) defined the eleven tiles required to draw knot mosaics. A tile may be empty, a rotation of a quarter arc, a horizontal or a vertical line, a rotation of a double arc, an overcrossing, or an undercrossing. Figure 1A shows the two crossing tiles.

An *n-mosaic* is an $n \times n$ array (or *board*) of mosaic tiles; $m \times n$ arrays are also allowed. A tile's *connection point* is where the tile's curve intersects the midpoint of an edge. A tile is *suitably connected* if each of its connection points touches a connection point of a contiguous neighbor tile. A *knot n-mosaic* is an *n-mosaic* where all tiles are suitably connected.

Methods to find knot mosaics include counting arguments (Hong et al. 2014) as well as writing properties by hand for specific types of knots on a 7×7 board and using computer scripts to fill in the combinatorics (Heap and LaCourt 2020).

Our Contribution

Our ultimate research goal is to enumerate all nontrivial knots on $m \times n$ boards. In contrast to the methods of the related work, our contribution is to use SAT solvers as an intermediate step to reduce the search space. We encode the knot mosaic constraints as a Boolean formula and use a SAT solver to find all satisfying assignments to the formula.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

SAT Formula Encoding

Boolean satisfiability (SAT) is the classic NP-complete problem. Since knot mosaics are a combinatorial problem, a SAT formula can encode suitably connected mosaics. Each variable corresponds to a tuple (i, j, k) , which means that tile k is at location (i, j) . The SAT solver sets every variable to either True or False based on the allowed tiles at neighboring locations. We encoded three increasingly more restrictive constraints. These constraints manage local properties of a mosaic, which are properties that relate to a tile and its immediate neighbors.

(a) **Basic encoding.** The board size is $m \times n$, infinite tiles are available, and there must be exactly one tile per board (i, j) location. We encoded the possible neighbors for each tile in the left, right, top, and bottom directions. We encoded the tiles that are allowed in the corners and edges (e.g., T9 may never be in a corner). The size of the corresponding SAT formula is $O(mn)$.

(b) **Require at least 3 crossing tiles.** We encode that at least 3 crossing tiles (T9 and T10) must be used and that those tiles must be 2 of one type and 1 of the other type. This simple improvement gives a better chance of producing a nontrivial knot (Figure 1C) but does not guarantee a nontrivial knot; for example, Figure 1D has three crossing tiles, but the knot is still the unknot. The size of the corresponding SAT formula is $O((mn)^3)$, which is large but still polynomial.

(c) **Ban trivial substructures.** We banned 2×2 trivial

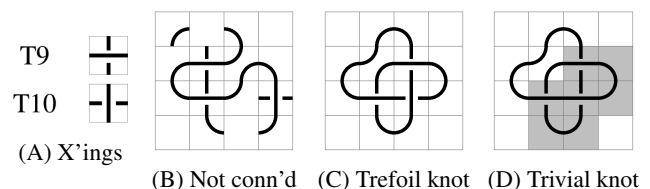


Figure 1: (A) The two possible crossing tiles T9 and T10. (B) A mosaic that is not suitably connected. (C) A representation of the trefoil knot, which is the simplest nontrivial knot. (D) A representation of the trivial unknot, which was formed by swapping exactly one crossing tile from the trefoil knot. Two 2×2 trivial caps are highlighted in gray.

m	n	added constraints	# vars	\sim # clauses	$\sim \varphi$ size (kB)	\sim time (sec)	# solutions	\sim improvement
4	6	(a) basic encoding	264	1,900	30	6,400	1,144,526	solver finished
4	6	(b) at least 3 T9/T10	264	2,400	130	600	437,332	3 \times
4	6	(c) ban trivial	264	5,700	190	<1	590	1940 \times
5	5	(a) basic encoding	275	2,000	30	90,400	4,183,954	solver finished
5	5	(b) at least 3 T9/T10	275	2,600	140	21,400	2,054,724	2 \times
5	5	(c) ban trivial	275	6,000	210	<1	1,374	3050 \times
5	6	(a) basic encoding	330	2,400	40	out of RAM	331,745,962	–
5	6	(b) at least 3 T9/T10	330	3,300	230	timeout	–	–
5	6	(c) ban trivial	330	7,600	320	6	49,570	6690 \times
6	4	(a) basic encoding	264	1,900	30	8,900	1,144,526	solver finished
6	4	(b) at least 3 T9/T10	264	2,400	130	1,100	437,332	3 \times
6	4	(c) ban trivial	264	5,700	190	<1	590	1940 \times
6	5	(a) basic encoding	330	2,400	40	timeout	331,745,962	–
6	5	(b) at least 3 T9/T10	330	3,300	230	timeout	–	–
6	5	(c) ban trivial	330	7,600	320	6	49,570	6690 \times
6	6	(a) basic encoding	396	2,900	50	out of RAM	101,393,411,126	–
6	6	(b) at least 3 T9/T10	396	4,100	390	out of RAM	–	–
6	6	(c) ban trivial	396	9,500	500	timeout	–	–

Table 1: Improvement when adding constraints, where *improvement* means “the integer factor by which the search space was reduced from the base case of all possible suitably connected knot mosaics to possibly nontrivial knots.” The SAT formula is φ . The time is the total running time for the solver `bc_minisat_all_static` (Toda and Soh 2016), which was run on a research computing cluster with Intel[®] Xeon[®] Gold 6150 CPU @ 2.70GHz. The timeout was 48 hours [172,800 seconds], and the RAM limit was 2048 MB. For entries where the solver finished the basic encoding, the number of mosaics agrees exactly with the results from (Hong et al. 2014); otherwise, the table gives the known count from Hong et al.

substructures such as flips, caps (e.g., the gray tiles in Figure 1D on the previous page), 4-tile circles, ellipses, etc. These trivial substructures are related to the standard *Reidemeister moves* (Reidemeister 1927). The size of the corresponding SAT formula is $O(mn)$ since for an $m \times n$ board, there are a fixed number of 2×2 trivial substructures to ban.

Results and Future Work

Table 1 shows our results.¹ As a self-check, the number of solutions found by the SAT solver for $m \times n$ and $n \times m$ boards agree exactly. Our selected constraints are effective in reducing the search space from intractable to tractable; for example, on a 6×5 board, the solver went from timing out to finishing in 6 seconds.

There are two interesting timing results for the basic encoding case [constraint (a)]. First, the running times for the 4×6 and 6×4 boards are 40% different. Second, the 5×6 board ran out of RAM, but the 6×5 board ran out of time. Since the only change was the order of m and n when writing the formula, for future work, we will study the SAT solver behavior on these instances.

The number of closed loops on a mosaic board is a global property of the mosaic. Because the current SAT formula encodes only local properties, the formula can produce links (two or more closed loops on the board). Since we are in-

¹We ran all possible combinations up to 6×6 boards, but to save space, we only report the board sizes with over 1 million knots.

terested in knots (a single closed loop) and not links, for the knot theory portion of our future work, we will encode the single-loop global property as a coloring in the SAT formula. For the SAT portion of our future work, we plan to investigate using parallel SAT techniques such as sharing the learned clause database among parallel cores.

Acknowledgments

We thank Ivona Bezáková and Angelina Brilliantova for their helpful comments. We thank RIT Research Computing for computational resources that contributed to our research results. Partially supported by NSF grants DUE-1819546 and DUE-1821459.

References

- Hall, D. S.; Ray, M. W.; Tiurev, K.; Ruokokoski, E.; Gheorghe, A. H.; and Möttönen, M. 2016. Tying quantum knots. *Nature Physics*, 12(5): 478–483.
- Heap, A.; and LaCourt, N. 2020. Space-efficient prime knot 7-mosaics. *Symmetry*, 12(4): 576.
- Hong, K.; Lee, H.; Lee, H. J.; and Oh, S. 2014. Small knot mosaics and partition matrices. *J. Phys. A*, 47(43): 435201.
- Lomonaco, S. J.; and Kauffman, L. H. 2008. Quantum knots and mosaics. *Quantum Inf. Process.*, 7(2): 85–115.
- Reidemeister, K. 1927. Elementare begründung der knotentheorie. In *Abh. Math. Semin. Univ. Hambg.*, volume 5, 24–32. Springer.
- Toda, T.; and Soh, T. 2016. Implementing efficient all solutions SAT solvers. *J. Exp. Algorithmics*, 21: 1–44.