Uniqueness in Puzzles and Puzzle Solving

David Eppstein

Minisymposium on Mathematical Puzzles and Games in
Theoretical Computer Science
ICIAM, Waseda University, August 2023

Uniqueness in mathematical puzzles

Trial and error (backtracking) works well for computers, but human puzzle solvers are generally encouraged to use deduction

This requires puzzles to be designed with unique solutions

3							5	8
	6	5						
	9		4					1
5			1			7		4
				7				6
							2	9
8		1	7				3	
	4		5	9				
					2			

A good sudoku puzzle

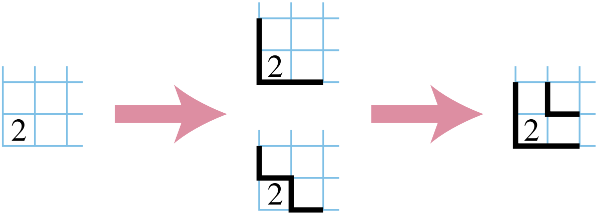
A bad sudoku puzzle

Knowledge of uniqueness can enable deductions

The “right hand rule” for finding the center of a maze
works when there is a unique path to the goal
can fail to work when the goal is surrounded by a loop of paths

Loopy / Slitherlink

Select edges from a grid (or other graph) to form a single cycle with correct number of edges around squares containing numeric clues



Uniqueness-based deduction rule: a 2 in a corner, surrounded by three blank squares, has two different (ambiguous) solutions. To prevent the ambiguity, some other part of the cycle must enter the square to block one of these solutions.

Unrelated to computational complexity

3-SAT instances with unique solutions are complete for NP under randomized Turing reductions [Valiant and Vazirani 1986]

Most of these puzzles have parsimonious reductions:
transformations from 3-SAT that preserve #solutions

The resulting instances have unique solutions and are computationally hard

Assuming uniqueness cannot make them easier

But, for deductive solving, uniqueness expands available deductions
$\Rightarrow$ uniqueness-based deduction makes some puzzles easier to solve

Later in the same sudoku

If 3's and 5's fill all colored squares, they could be swapped

3							5	8
	6	5			7		9	3
2	9	8	4	■	■	6	7	1
5	3		1			7	8	4
	8			7	■	■	1	6
				■		■	2	9
8	2	1	7			9	3	5
7	4	3	5	9			6	2
	5				2		4	7

Only way to escape ambiguity is to place one of the 3's in the right column of the center 3×3 block

Without using ambiguity, a deductive sudoku solver that I wrote was unable to solve this without resorting to non-local 2SAT-based deductions

Monotonicity

Consider

- Puzzles whose solution consists of local pieces of information, like loopy & sudoku
- Deductions that look for "trigger" patterns in the neighborhood of a piece, and set the piece when they find a trigger
- Monotone: if a trigger pattern is present, then it will remain present after setting any piece

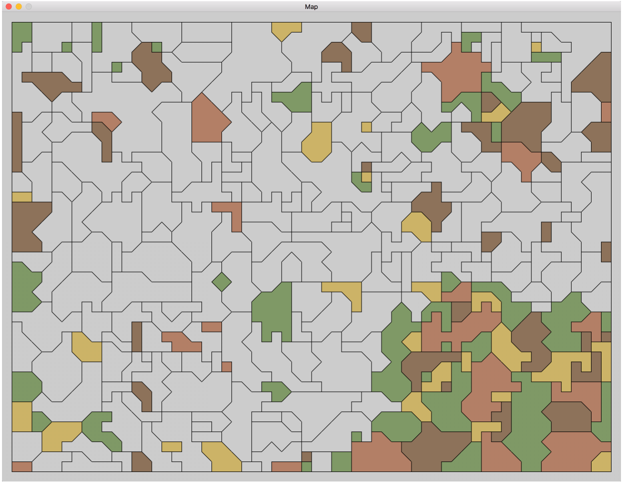
  (Intuition: if deduction was valid without knowing how to set the piece, it should remain valid no matter how we set it)

Then rule order doesn't matter: once some setting of pieces is deducible, it remains deducible until all its pieces have been set

Mathematically, this means that the orderings in which you can fill in the pieces using these rules form an antimatroid

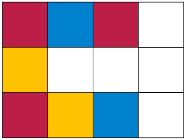
Map

From Simon Tatham's puzzle collection:
complete a partial 4-coloring of a map ("precoloring extension")

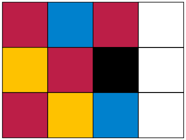


Uniqueness-based rules can be non-monotonic!

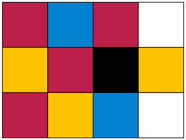
Suppose we see this as part of a larger puzzle
(white squares = not yet colored)



If inner white square were surrounded by yellow & blue it would be ambiguous ⇒ its remaining neighbor must be black



To force neighboring square to be black, its outer neighbor must be yellow



But if we fill in the black square, and treat it as a given rather than remembering that it must be forced to be black, we cannot make the final deduction! (⇒ complicated knowledge representation issues)

Why is non-monotonicity bad?

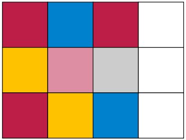
Monotonic $\Rightarrow$ safe to choose any rule that gets triggered.

Non-monotonic $\Rightarrow$ lost in a maze of inequivalent rule orderings

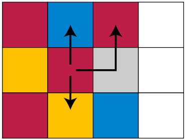
Monotonicity from weakened uniqueness

Instead of assuming the solution to be unique, assume it to be anchored: every Kempe chain (maximal two-colored subset of regions) includes at least one clue

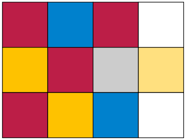
When making a deduction, shade newly-colored region lightly (it is not yet anchored)



If a lightly-shaded region has two-color paths to anchored points, for all other colors, shade it as dark (it has now been anchored)



Now we can deduce that every light-colored region needs neighbors of all other colors



Summary

Uniqueness doesn't affect computational complexity but can be very important in human deduction (or its computational simulation)

Non-unique rules are often monotonic $\Rightarrow$ rule ordering is trivial

Uniqueness leads to complex knowledge states (e.g. some later choice must force this part of the puzzle to behave this way) of potentially unbounded complexity, and unclear choice of rule order

For the map puzzle, we can regain monotonicity through anchoring rather than uniqueness, but this restricts our deductions

What about other puzzles like sudoku and loopy?

References and image credits

Elliott Brown. A look around the gardens at Château de Cormatin. CC-BY-SA image, June 5 2017. URL https://commons.wikimedia.org/wiki/File:Ch%C3%A2teau_de_Cormatin_-_the_gardens_-_boxwood_maze_(35775073455).jpg.

L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1):85–93, 1986. doi: 10.1016/0304-3975(86)90135-0.