CSCI 1710 Capstone: Modeling Sudoku Board Properties and Uniqueness

Christopher Pellinger
Faculty Sponsor: Tim Nelson

Background

The goal in this project is to determine properties of incomplete sudoku boards with unique solutions. In order to determine this, we had to model a sudoku board, find a way to generate valid full boards (and determine validity), remove values from those boards to end up with a solvable partial board, and finally ensure that the partial board only had one possible solution. We began by working in Froglet and then Temporal Forge, but we eventually switched to Python and Z3 for our final model.

Model

Multiple models were created throughout the process of the project. The first model was created in temporal Forge. This model has the ability to create incomplete sudoku boards, but not with unique solutions. This model is structured with a partial function that represents the places on the board. To create an incomplete board, first a complete board is made and then cells are removed literally until a number missing from each 3x3 is reached. This process was visualized using a custom visualization built in javascript. It can produce outs below.

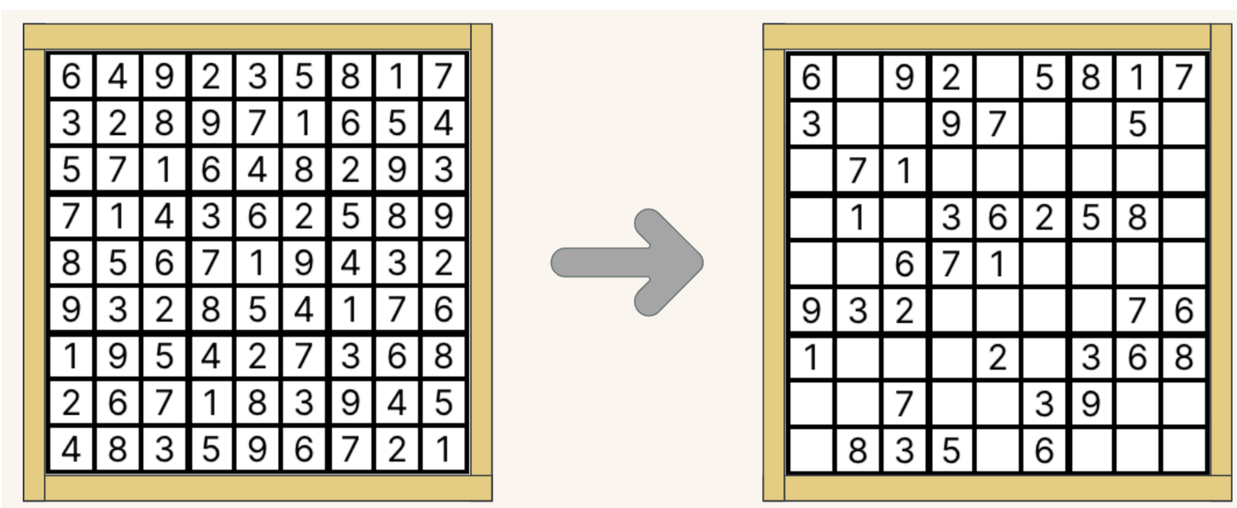

Figure 1. Forge Outputs using a Custom Visualization

Forge some limitations though imposed by a bit width of 4. This prevented the use of numbers 8 and 9 meaning the model had to use numbers -1 through 7. This is corrected by the visualizer. In addition this meant it was hard to specify large quantities of clues to be removed in the board. Forge also could not generate incomplete boards. In order to generate complete boards, all possible solutions must be able to be generated and compared. Forge did not have this ability. For this reason, the model was made in python with the use of Z3.

Z3 allowed for the creation of incomplete sudoku boards with a unique solution. In this model the process is still started by generating a complete sudoku board. The model then iteratively removes X number of cells or "clues." After each removal the model attempts to solve with the constraint that it cannot be the original board. If there are no other solutions the model continues removing cells. This allowed for the creation of incomplete boards with unique solutions to test various properties.

Findings

We found that when generating sudoku boards there are a specific number of cells that can be removed. All solution boards will be able to be reduced by 55 clues and will always contain a unique solution. When removing 56 to 64 clues some complete boards will not contain a unique solution but others will. When attempting to remove >64 clues all sudoku boards will not have unique solutions. Another property tested was pair diagonals. This is displayed below. The property is that if a solution board contains this type of pattern one of the 4 numbers must remain when removing cells. This property did not hold when testing. The testing used in this project was in python with the use of the hypothesis library.

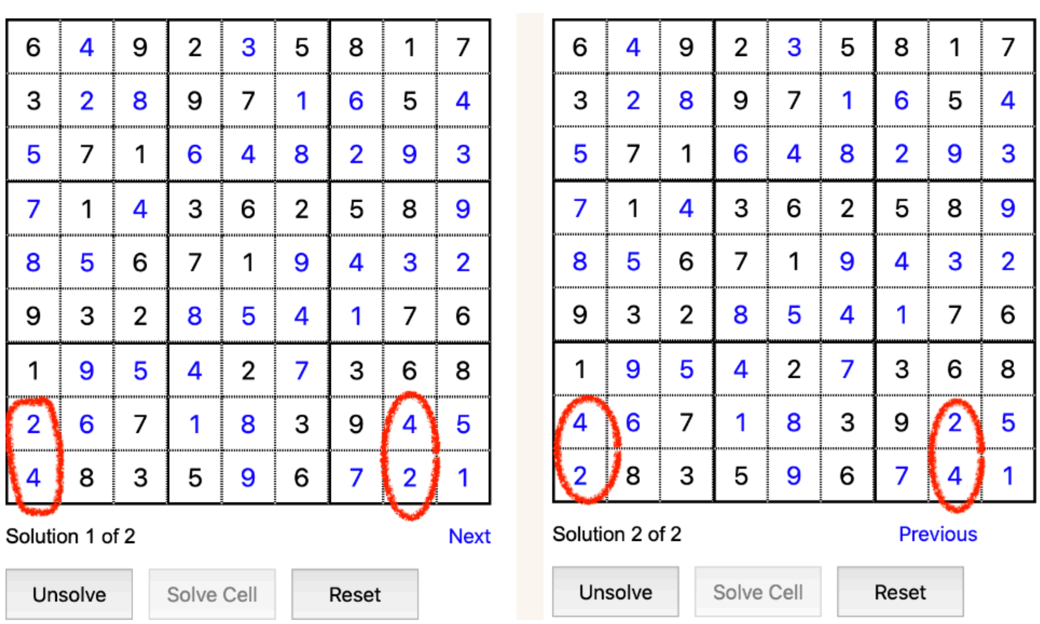

Figure 2. Pair Diagonals

This model can further be used to test other properties of incomplete boards with unique solutions. These properties could be useful in further optimizing sudoku puzzle creation.