

Sudoku SAT Reduction

Rob Bishop, Caleb Graves, Vineel Nagisetty

Overview

As an NP-Complete problem, the game of sudoku can be reduced to a SAT solvable state by formalizing the rules into a series of conjunctive statements. We will present the process of this reduction here with the intention of creating a CNF file that may be used in a commercial SAT solver.

Sudoku rules primer

The rules of sudoku are explained here for completeness. The game is played on an $N \times N$ board where N is a prime number. Here N will be referred to as the order of the board, such that a 4×4 board is a board of order 4. Note that while the canonical version of Sudoku uses a board of order 9, the problem can be abstracted to arbitrarily sized boards.

For the board to be completed in a winning state the following must all apply:

- The value v of any number in a cell on the board is an integer $\in \{1, N\}$
- Each cell of the board must contain a single value
- Each row r of the board must contain one, and only one instance of each number $\in \{1, N\}$
- Each column c of the board must contain one, and only one instance of each number $\in \{1, N\}$
- Each subsquare s of the board must contain one, and only one instance of each number $\in \{1, N\}$

CNF Interpretation of Sudoku Rules

Here, we will use the notation $(rcv)|r, c, v \in [1, N]$ to represent the location and value of a cell. For example: (123) represents a value of 3 in the cell at position (1, 2).

To formalize the rules we will adopt the nomenclature from Kwon & Jain[?] and break the cell, row, column, and subsquare rules into two separate constraints:

definedness: each cell, row, column and subsquare must contain one number

uniqueness: each cell, row, column and subsquare must contain no duplicate numbers

- Definedness Constraints:

For cells, definedness requires listing each possible permutation of values for each cell:

$(000) \vee (001) \vee (002) \dots \vee (00N) \wedge$

$(010) \vee (011) \vee (012) \dots \vee (01N) \wedge$

...

For rows, columns, and subsquares definedness is performed by fixing the values of the other variables and generating each possible permutation. For example, to represent definedness of a row we iterate through each permutation of c, v and define a clause for each cell within the row:

$(000) \vee (010) \vee (020) \dots \vee (0N0) \wedge$

$(001) \vee (011) \vee (021) \dots \vee (0N1) \wedge$

...

- **Uniqueness Constraints:**

Representing uniqueness in CNF is done by conjoining every possible permutation of duplicate values as negations. For cells, for example we use the form $(rcv_1) \vee (rcv_2)$ for each possible permutation of v_1, v_2 :

$(000) \vee (001) \wedge$

$(000) \vee (002) \wedge$

...

This same approach holds for rows, column and subsquares. Though defining iteration through subsquares is mathematically somewhat more complex, the principle of fixing the other values and iterating through each permutation remains the same.

Defining the Playing Field

Typical sudoku puzzles begin with a board of mostly empty squares with several values already filled in. To represent these pre-defined squares in the above form, you simply append a single (rcv) representing each filled cell. Doing so will allow a traditional SAT solver to determine if the board has a solution (obviously all commercial puzzles have a solution, but the reduction presented here will determine the satisfiability of any arbitrarily filled board).

Conclusion and Performance

Using the above method it is fairly simple to iteratively create a CNF definition of the rules of sudoku such that any boards satisfiability can be determined. It is worth noting that much of the modern literature on the topic is dedicated to finding more efficient implementations as the above method generates much redundancy. In practice, each defined cell effectively reduces the possibility space a great amount, and several of the iteratively generated clauses are redundant with eachother.

References

- [1] Gihwon Kwon, Himanshu Jain. *Optimized CNF Encoding for Sudoku Puzzles*. 2009