Sudoku and Other Related Problems

E. Routledge

01 Nov 2022

1 Sudoku is hard

Sudoku is a simple logic game, in the standard 9×9 (or $3 \times 3 \times 3 \times 3$) one must complete the grid such that every row, column and box contains the numbers 1 to 9, that is all. So, yes, sudoku is simple in the sense it can be described in a single rule but it is in fact a hard problem to solve. Instances of the puzzle requiring complex x wing and y wing strategies are not what makes the puzzle hard to solve, it is hardness through a provable, mathematical (computational) lense for which this paper cares.

1.1 Finding a solution

The standard way to solve a 9×9 sudoku puzzle is by the backtracking algorithm. This is a brute force method with a few optimisations. One can expect to find this algorithm in a computer science course introduction to recursion, that is to say it is not a complex concept and while useful for the usual sizes, as soon as we increase to 16×16 this becomes infeesible.

Listing 1: Backtracking

```
def Backtracking(grid, dimension):
    for i in range(dimension):
        for j in range(dimension):
            if grid[i][j] is empty:
                for k in range(dimension):
                    grid[i][j] = k
                    if valid(grid):
                        result = Backtracking(grid,dimension)
                            if result is a grid:
                                return result
                return False
```

```
return grid
```

Why does brute force not work for larger examples?

Sudoku is NP-complete that is to say it is computationally hard and no known algorithm can solve this in resonable time (polynomial).

Proof

We do this through a reduction to a known np-complete problem. Assume we have an oracle Φ that when given an instance of the problem (in this case a grid of numbers and blanks) will, in polynomial time or less return True if it can be solved and False otherwise.

$$\Phi(\text{grid}) = \begin{cases} \text{True if a solution exists} \\ \text{False if a solution does not exist} \end{cases} \tag{1}$$

1.2 Uniqueness is hard?

It is hard to determine if a puzzle has a unique solution?

- generate uniqueness