Oussama Oulkaid

SAT/SMT Solving
Sudoku Solving

January, 2022

This document is a description of the two Sudoku solvers built for this project. The first part details the Z3 theorem prover based solution, whilst the second details the implementation of a chronological backtracking based algorithm.

1 Z3 based solver

Our goal here is to use the Z3 API in Python to solve a given Sudoku problem. For this purpose, we create a solver object and add to it the following clauses:

- Distinct values in each line
- Distinct values in each column
- sol values must be natural numbers contained within the interval [1,n]
- Specify initial values of the sudoku problem as defined in the grid representing the problem
- Ensure uniqueness of values in each box

Then, we check the satisfiablity of the resulting formula and print the solution.

2 Backtracking based search engine

The following is the algorithm.

```
Input: Sudoku problem
Result: Sudoku solution
Fill in the trivial cells in the grid of size n
pos := ∅;
pre := ∅;
i := 0;
j := 0;
return FIND_SOLUTION(grid, n, i, j, pos, pre)
```

```
function FIND_SOLUTION(grid, n, i, j, pos, pre)
     if grid[i][j] == 0 then
    if cell encountered for the first time then
        pre.add(i,j) get a valid digit for this cell
        if no digit is possible then
            tofo
        end
    end
     end
end function
```

Algorithm 1: Backtracking algorithm based Sudoku solver

Comparison

TODO: express the downsides and upsides of my implementation. Have some graph to sythesize some benchmarking testing.