Université de Liège

Logic for Computer Science

Sudoku solving using propositional logic

Author:
Iván Pérez Longa

November 2022

1 Introduction

Since its creation around the year 1970 and its popularization in the 80's, Sudoku is a world-renowned enigma with an infinite number of variants and modifications. In this project, we are proposed to develop a program that generates and solves these mathematical enigmas of different dimensions through logic. In the following points of this essay, various key concepts in solving this problem will be discussed.

2 Development

To have a clear idea in the development of this project, and better understand the problem that concerns us, numerous documentation has been required. Among them stand out Sat Solving, Sudoku as SAT, Sudoku as a Constraint problem or Sudoku as a SAT problem. Also mention the help of one of my colleagues, Pierre Gaillot, who thanks to a discussion with him on the subject made me better understand the encoding of values in solving sudoku puzzles with a dimension greater than 9.

2.1 Constraints

Taking into account that the main topic of this project is logic and not programming, I am going to focus this short report on the constraints that I have used to generate and solve the sudoku puzzles, trusting that with the numerous comments that I have written in the code this is perfectly readable and understandable.

2.1.1 Resolution Constraints

- At most one number in each entry: $\wedge_{row=1}^{N+1}\wedge_{column=1}^{N+1}\wedge_{value=1}^{N}\wedge_{auxval=value+1}^{N+1}(\neg s_{row,column,value}\vee \neg s_{row,column,auxvalue})$
- Each number at least once per row: $\wedge_{column=1}^{N+1}\wedge_{value=1}^{N+1}\vee_{row=1}^{N+1} s_{row,column,value}$
- Each number at least once per column: $\wedge_{row=1}^{N+1}\wedge_{value=1}^{N+1}\vee_{column=1}^{N+1} s_{row,column,value}$
- Each number at least one per sub-grid: $\vee_{value=1}^{N+1}\wedge_{plusrow=0}^{n}\wedge_{pluscolumn=0}^{n}\wedge_{row=1}^{n+1}\wedge_{column=1}^{n+1} s_{(n*plusrow+row)(n*pluscolumn+column)value}$
- And the constraint to include the given clues.

2.1.2 Another Solution Constraint

- Negate current solution: $\wedge_{column=1}^{N+1}\vee_{row=1}^{N+1}(\neg s_{row,column,sudokusol[row-1][column-1]}|$ if it is not a clue)

3 Improvements

Despite all that has been said, the model that has been developed is far from ideal, since it takes its time to generate large sudoku puzzles. As I have already said in the previous point, the program can solve, verify if there is more than one solution, and generate sudoku puzzles of size 4, 9, 16, and 25, but it is difficult for it in large dimensions. In my opinion the code is not perfect, but as far as I understand, where a significant improvement could be made is in the internal logic of the program, making it not generate so many redundant clauses. This should have a big impact on the running time of the program, and would be my first topic if I had more development time.

4 Conclusion

As a conclusion for this essay, I would like to point out that this project has been one of my first single Python developments, and it has helped me tremendously to better understand the language and feel more comfortable with it. I am happy, then, with the final result of the project and with the knowledge acquired.

Code in https://github.com/ivanperezlonga/LogicSudokuSAT.git