A Brief Note on SAT Solvers and Sudoku

Introduction

SAT solvers are programs designed to determine whether a logical formula can be satisfied by some assignment of truth values. They are widely used in computer science for tasks such as verification, planning, and constraint solving. One of the foundational approaches behind many SAT solvers is the DPLL algorithm, which systematically searches for a satisfying assignment. Modern solvers build on these ideas while remaining efficient in practice.

Conjunctive Normal Form

A formula is in conjunctive normal form (CNF) if it is written as a conjunction of clauses, where each clause is a disjunction of literals. This standardized structure allows SAT solvers to process formulas efficiently.

An example of a formula in CNF is $(x_1 \vee x_2) \wedge (x_3 \vee x_4 \vee x_5)$.

Sudoku Solving Requirements

A valid Sudoku solution requires that each cell contains exactly one number from 1 to 9. Additionally, every number must appear exactly once in each row, each column, and each 3×3 subgrid. We can find a solution to a sudoku by encoding each of the previous conditions as CNF formulae and performing the DPLL algorithm on them; the resulting truth values of the variables involved should give a valid solved sudoku.