

SAT Assignment Part 1: Encoding Non-Consecutive Sudoku to CNF

Andrzej Szczepura, Patrick Koopmann

31/10/2025

Overview

For this and the next assignment, you will be writing an SAT solver, which solves sudoku puzzles. We will be working on a variant called “Non-Consecutive Sudoku” — all the standard sudoku rules apply, with an extra constraint — **orthogonally adjacent cells cannot differ by 1**.

Write a program that reads a Sudoku puzzle (sizes: 9×9 , 16×16 , or 25×25) and outputs a DIMACS CNF encoding which corresponds to the Non-Consecutive Sudoku rules (cell, row, column, box, non-consecutive rule). Your encoding should include the initial clues of a given puzzle as well, encoded as unit clauses.

Input Format

A text file with an $N \times N$ grid of integers (space-separated). The value \emptyset denotes an empty cell. N is a perfect square (e.g., $N \in \{9, 16, 25\}$).

We have provided you with an example 9×9 , 16×16 and 25×25 puzzle in this format.

Output Format (DIMACS CNF)

The program must produce a DIMACS CNF file:

```
p cnf <num_vars> <num_clauses>
<lit1> <lit2> ... 0
...
```

Do not include trailing \emptyset s inside your clause lists in code; the writer appends \emptyset per line.

Variable Mapping (required)

Use the following mapping exactly:

$$\text{var}(r, c, v) = r \cdot N^2 + c \cdot N + v,$$

where $r, c \in \{0, \dots, N - 1\}$ (0-based indices) and $v \in \{1, \dots, N\}$.

Constraints to Encode

Let $B = \sqrt{N}$ (so each box is $B \times B$). Encode the following:

(1) **Exactly one value per cell.**

For each cell (r, c) , exactly one $v \in \{1, \dots, N\}$ is true.

(2) **Row constraint:**

For each value v and each row r , exactly one column c has v .

(3) **Column constraint:**

For each value v and each column c , exactly one row r has v .

(4) **Box constraint:**

For each value v and each $B \times B$ box, exactly one cell in that box has v .

(5) **Non-consecutive rule:**

For every cell (r, c) and each orthogonal neighbor (r', c') :

$$|\text{value}(r, c) - \text{value}(r', c')| \neq 1.$$

(6) **Clues:**

Unit clauses corresponding to a given puzzle — for a given digit $v > 0$ at (r, c) , add the unit clause $X_{r,c,v}$.

Command-Line Interface

Your program must implement:

```
python main.py --in <puzzle.txt> --out <instance.cnf>
```

The provided starter includes `main.py` and an empty `encoder.py` stub. Implement the function `to_cnf(input_path)` in `encoder.py` to read the puzzle and produce the clauses and the variable count.

Submission

Submit a ZIP archive named:

`group_<number>_a1.zip`

with the following top-level files:

- `main.py`
- `encoder.py`

Deadline: 23:59 07-11-2025

Tips

- Implement a helper for “Exactly One”: one clause for “at least one”, and pairwise clauses for “at most one”.
- Be careful with indices: v ranges $1..N$, while r, c are $0..N - 1$.