# Project 1: SAT Encoding of a 3 × 3 Sliding Puzzle

## Task Description

Generate a bounded planning encoding in SAT that solves a sliding puzzle of size 3 × 3. The field contains eight stones numbered with $1 - 8$. One field is empty. The stones on the fields left, right, top, or bottom of the empty field can be moved on the empty field. The goal is to sort the stones in ascending order such that the empty field is in the lower right corner. Below an possible initial state is shown on the left and the goal state is shown on the right.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 8 | 7 | 4 |
| 6 | 5 | |

$\Rightarrow$

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | |

### Tasks

Implement a tool that generates a propositional formula using the planning encoding of the lecture. In particular, your encoding should include:

- A given initial state. The initial state is given as an integer array of size 9. The left board of the example above is represented as `[1, 2, 3, 8, 7, 4, 6, 5, 0]` where `0` stands for the empty field. No user interface is required.
- A global variable that allows to set the number of steps.
- The goal state which is `[1, 2, 3, 4, 5, 6, 7, 8, 0]`.
- The definition of the moves.
- All the necessary constraints (at-most-one move, frame axioms).
- Generate code for the Limboole SAT solver (http://fmv.jku.at/limboole).

## Submission Instructions

- The task can be solved in groups of two.
- In Moodle, one submission is sufficient, but in the submission clearly provide both names.
- In the week after the submission, you will be invited to orally present your solution (via Zoom or in the office of Martina Seidl).
- Also for meaningful partial solutions you will get points.
- Submit your code that generates the propositional formula for Limboole.

## Hints

- The task can be solved directly with the planning encoding presented in the lecture.
- Start with identifying the necessary propositional variables. Think about good naming conventions. E.g., `mv_t_s_xy` could stand for moving stone `s` from field `x` to field `y` at time `t`. Define functions that generate the names (so typos can be avoided), e.g., `mk_mv (time, field, src, trg)`.
- The program to generate the encoding is not complicated if you design it properly. However, the debugging might be a bit challenging. For this purpose, it is easier to start with problems that require only one or two steps for their solution.
- The encoding will have many variables. Find a way to filter out the relevant variables, e.g. with the command grep.

```
python3 puzzle.py | limboole -s | grep "= 1" | grep mv
mv_0_5_HI = 1
mv_1_6_GH = 1
mv_2_8_DG = 1
mv_3_7_ED = 1
...
```