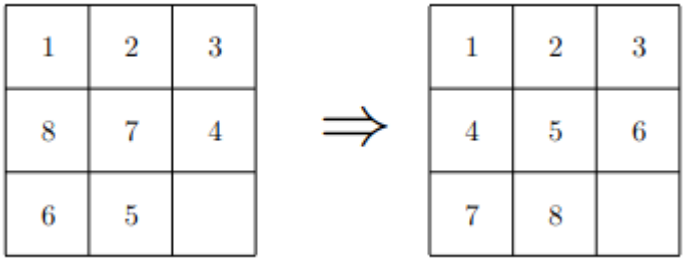
Project 1

Nikita Fordui



Field is numbered 1 to 9, left to right, up to bottom.
Empty field corresponds to value 0, others - to value 1 through 8.
Final program has two inputs - field in form of list of numbers and another integer - length of the solution. To find a solution of arbitrary length you can just test out different lengths starting from n = 1 of solution one by one until you get the solution from the limboole.

State variables:

For each field - 9 variables which encode which of the values 0 through 8 is on the field.
Naming - s **t** *f* ***i*** n**j**, where i - field number, j - stone number, t - number of state (time).
In total 81 state variables.

Action variables:

All valid pairs of fields from/to is corrensponding to 8 moves - one for each stone.
In total 8 * 24 = 192 moves and 192 action variables.
Naming - mv**t** *s**j*** p **ip ipp**, where j - stone number (1-8, doesn't include 0 stone), ip and ipp - positions to and from pairs of possible moves (see constant POSSIBLE_MOVES in code).
End state would be [1, 2, 3, 4, 5, 6, 7, 8, 0]

Action definition

Action definition in code of the program is generated from generate_action_changes function. Each of the generated action definition formulas follows structure (action → (prerequisites & effects)).

Action is just a variable mv**t** *s**j*** p **ip ipp**.

Prerequisites is a cube containing two literals without negation: one corresponding to the fact that starting position for move should have a stone we want to move - s**t** ***f ip*** n **j**, another - corresponding to the fact that destination has no stone (empty stone zero). Prerequisites also contain literals with negation corresponding to two facts: that destination field shouldn't have any other stone except from one you move and that target field shouldn't have any stones except zero stone.

Effects is a cube which contains two literals corresponding to change of location of zero stone and the stone you are moving and literals corresponding to the fact that now destination and target doesn't have stones they had on previous turn.

Frame axioms

For the frame axioms part, I need to mention that state variables for zero "empty" stone and all other stones are treated differently.

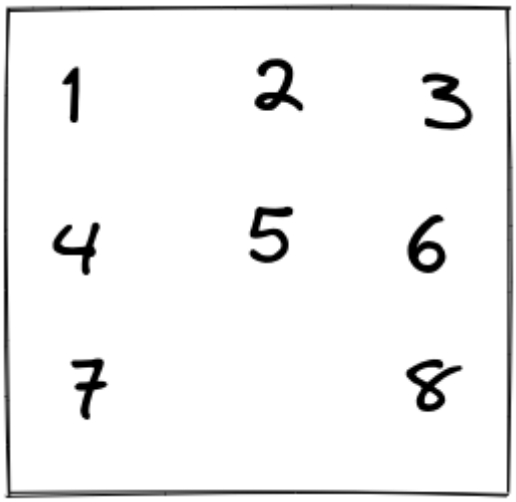
To change state with zero stone (make zero stone disappear) we need to perform one of the actions which moves any other stone into the position of the zero stone's, with any other stone it's vice versa - we need the stone to be moved to any other position. With symmetrical frame axiom (stone appears on the field) actions needed for the frame axiom for zero are those which move any stone from the current field, for any other stone - move this stone out of the field.

Other constraints

All other constraints are pretty straightforward and are documented in the code itself.

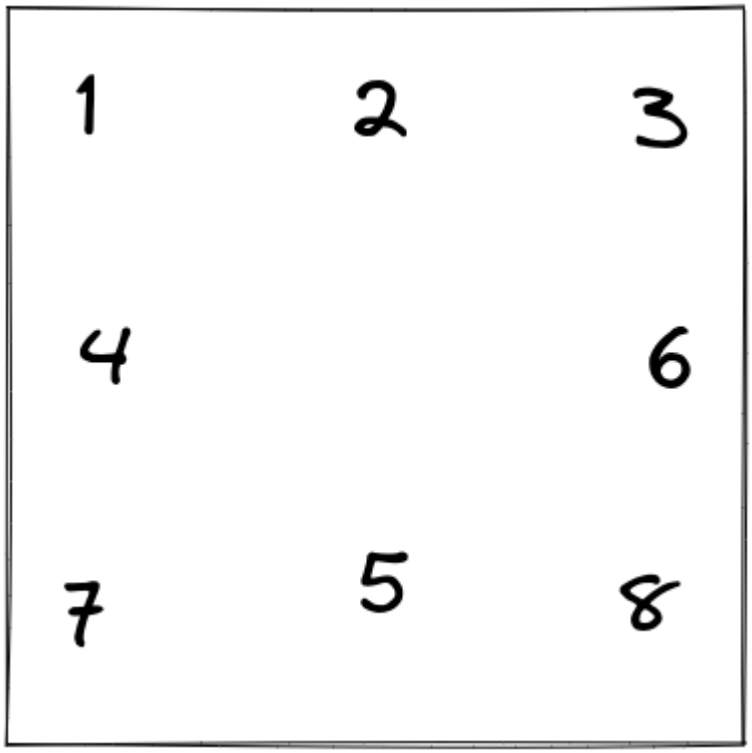
Examples

One-step



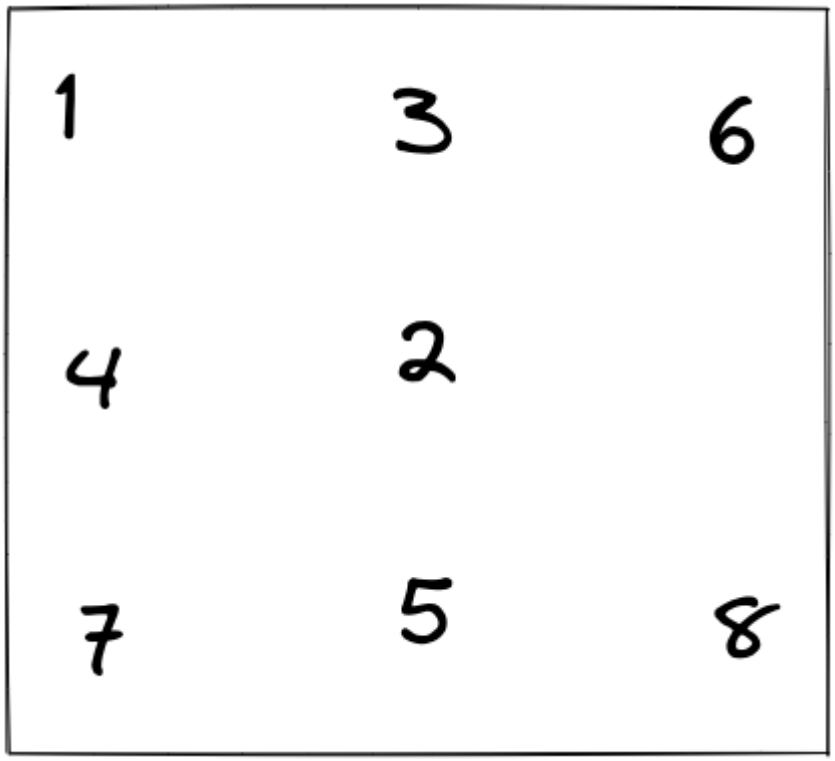

```
[puzzle_sat_encoding] python main.py "[1, 2, 3, 4, 5, 6, 7, 0, 8]" 1 | limboole | grep mv
| grep "= 1"
mv0_s8_p98 = 1
```

Two-step




```
[puzzle_sat_encoding] python main.py "[1, 2, 3, 4, 0, 6, 7, 5, 8]" 2 | limboole | grep mv
| grep "= 1"
mv0_s5_p85 = 1
mv1_s8_p98 = 1
```

Five-step




```
[puzzle_sat_encoding] python main.py "[1, 3, 6, 4, 2, 0, 7, 5, 8]" 5 | limboole | grep mv
| grep "= 1"
mv0_s6_p36 = 1
mv1_s3_p23 = 1
mv2_s2_p52 = 1
mv3_s5_p85 = 1
mv4_s8_p98 = 1
```