FRANCESCO ANTICI

Z3 SAT EXERCISES THEORY

KNIGHTS AND KNAVES

- There is an island in which certain inhabitants, called knights, always tell the truth, and the others, called knaves, always lie. It is assumed that every inhabitant of this island is either a knight or a knave.
- Suppose an inhabitant A says: “Either I am a knave or A is a knight.” What are A and B?

VARIABLES

- We should consider that both a and b can be either a knight or a knave but nothing else, so if a_{knight} holds, a_{knave} is necessarily false, $a_{knight} = \neg a_{knave}$.
- We can represent these facts simply defining two boolean variables, one for each individual, representing if he is a knight (or alternatively a knave).

CONSTRAINTS

- From what a states, we can conclude that: $a_{knave} \vee b_{knight}$.
- If a is a knight, then a's statement is true:

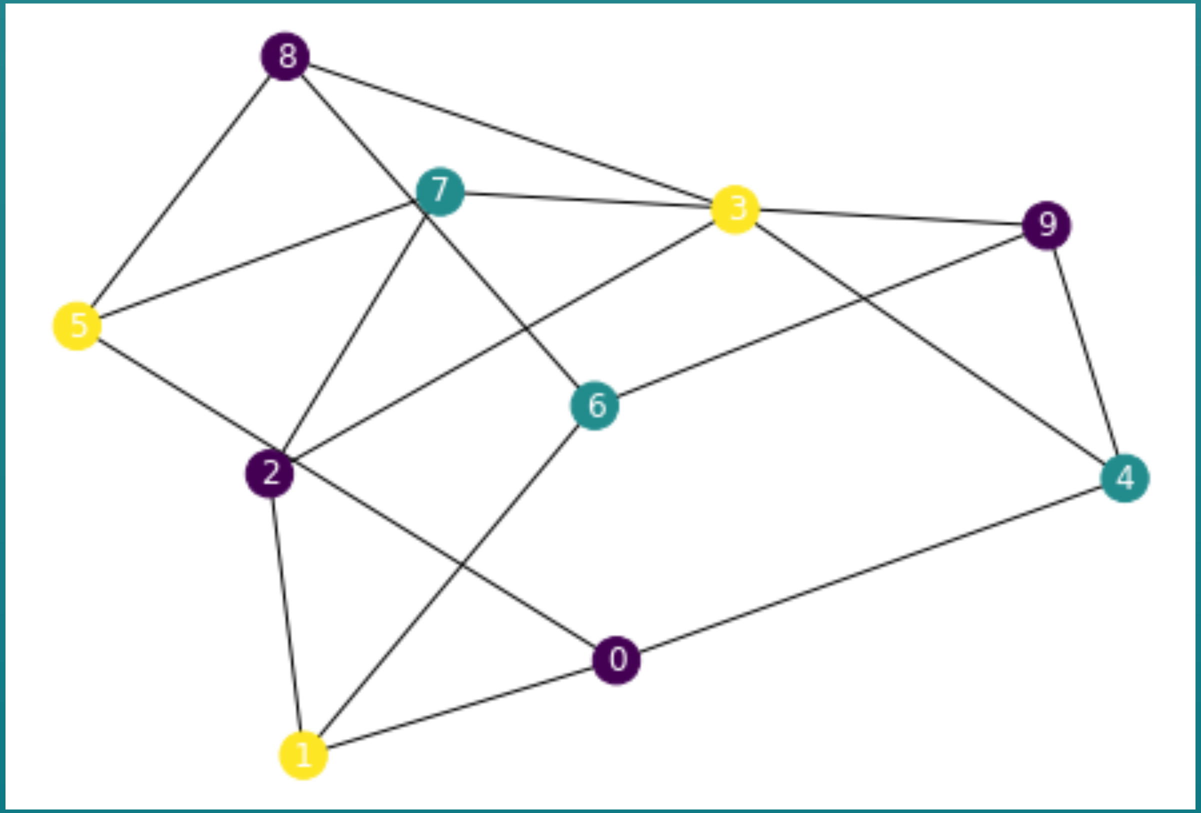
$$a_{knight} \rightarrow (a_{knave} \vee b_{knight})$$

- If a is a knave, then a's statement is a lie:

$$a_{knave} \rightarrow \neg(a_{knave} \vee b_{knight})$$

COLORING GRAPH PROBLEM

- Another famous exercise is the coloring graph problem.
- Given a graph $(v_1, \ldots, v_n, E)$ and d colors, we need to assign a color to each vertex, s.t. if $(v_i, v_j) \in E$ then color of v_i is different from color of v_j.



VARIABLES

- We have to represent the assignment of a color to a vertex, so we can use nxd boolean variables $v_{i,j}$ with $i \in \{1,..,n\}$, and $j \in \{1,..,d\}$.
- $v_{i,j}$ is true if vertex i is colored with color j, false otherwise.

CONSTRAINTS

- Each vertex has at least one color:

$$\bigwedge_{1 \leq i \leq n} \bigvee_{1 \leq j \leq d} v_{i,j}$$

- Each vertex can have at most one color as well:

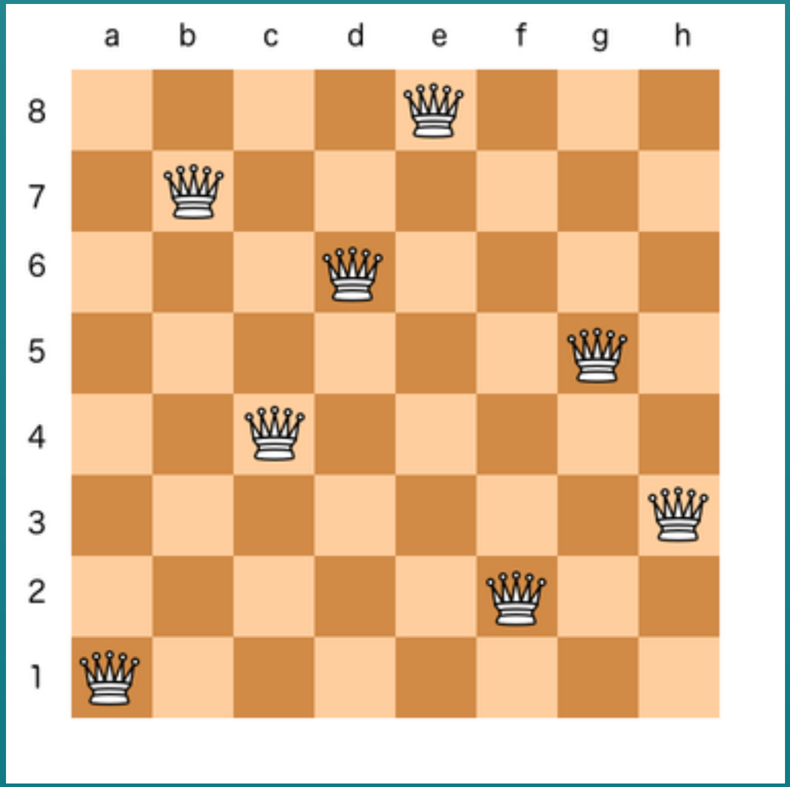
$$\bigwedge_{1 \leq i \leq n} \bigwedge_{0 < j < k \leq d} \neg (v_{i,j} \wedge v_{i,k})$$

- Each edge must have different colors in its vertices:

$$\bigwedge_{(v_i, v_j) \in E} \bigwedge_{1 \leq k \leq d} \neg v_{i,k} \vee \neg v_{j,k}$$

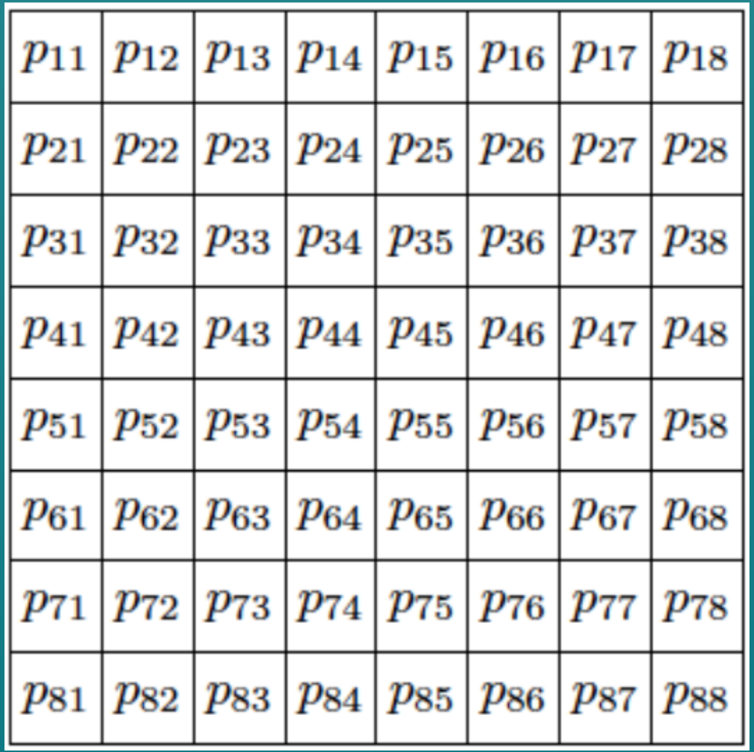
N-QUEENS

- Placing n chess queens in a nxn chessboard so that no queens threatens each other is called the n-queens problem.
- The solution requires that no two queens share the same row, column or diagonal



VARIABLES

- We have to represent the position of the queen in the chessboard, so we can use nxn boolean variables $p_{i,j}$ with $i, j \in \{1,..,n\}$.
- Each variable is true if a queen is in that particular position.



p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}
p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{26}	p_{27}	p_{28}
p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}	p_{38}
p_{41}	p_{42}	p_{43}	p_{44}	p_{45}	p_{46}	p_{47}	p_{48}
p_{51}	p_{52}	p_{53}	p_{54}	p_{55}	p_{56}	p_{57}	p_{58}
p_{61}	p_{62}	p_{63}	p_{64}	p_{65}	p_{66}	p_{67}	p_{68}
p_{71}	p_{72}	p_{73}	p_{74}	p_{75}	p_{76}	p_{77}	p_{78}
p_{81}	p_{82}	p_{83}	p_{84}	p_{85}	p_{86}	p_{87}	p_{88}

CONSTRAINTS

- At least one queen on each row and column:

$$\bigwedge_{1 \le i \le n} \bigvee_{1 \le j \le n} p_{i,j} \qquad \bigwedge_{1 \le j \le n} \bigvee_{1 \le i \le n} p_{i,j}$$

- At most one queen in each row and column:

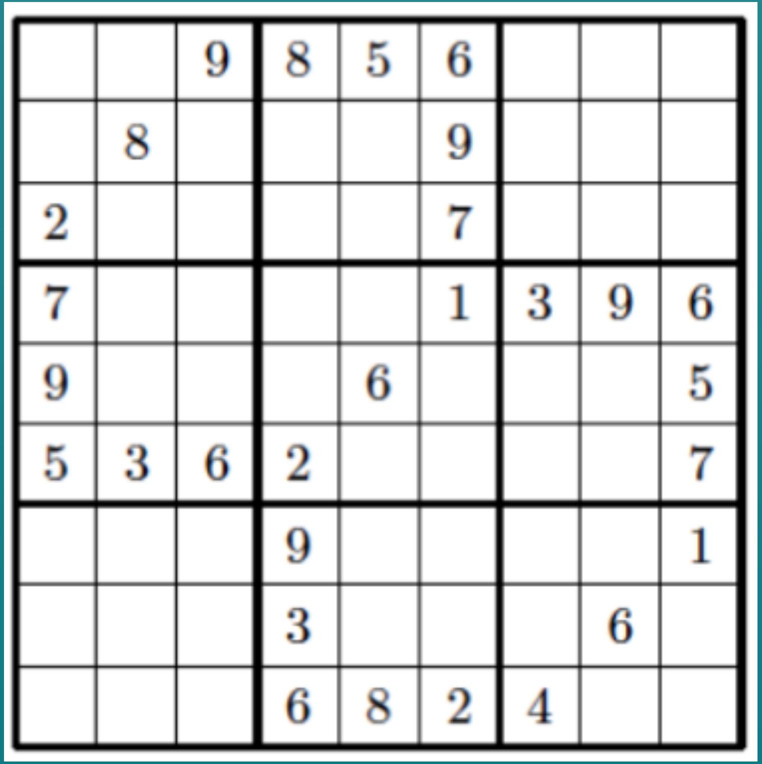
$$\bigwedge_{1 \le i \le n} \bigwedge_{0 < j < k \le d} \neg(p_{i,j} \wedge p_{i,k}) \qquad \bigwedge_{1 \le j \le n} \bigwedge_{0 < i < k \le d} \neg(p_{i,j} \wedge p_{k,j})$$

- At most one queen in each diagonal:

$$\bigwedge_{1 \le i < i' \le n} \bigwedge_{j,j' : i+j = i'+j' \vee i-j = i'-j'} \neg(p_{i,j} \wedge p_{i',j'})$$

SUDOKU

- Sudoku is a logic-based, combinatorial number-placement puzzle.
- In classic sudoku, the objective is to fill a 9 × 9 grid with digits so that each column, each row, and each of the nine 3 × 3 sub-grids contain all of the digits from 1 to 9.



		9	8	5	6			
	8				9			
2					7			
7					1	3	9	6
9				6				5
5	3	6	2					7
			9					1
			3				6	
			6	8	2	4		

VARIABLES

- We have to represent the position of the cell in the grid and the number associated to it.
- We need 9x9 variables for the position and other 9 each to fix the number, the total number of variables is 9x9x9.

$$v_{i,j,k} \qquad \forall i, j, k \in \{1,..9\}$$

CONSTRAINTS

- For this instance is important to define the $exactly\_one$ constraint, which, given V a set of boolean variables, can be defined as:

$$exactly\_one(V) \equiv at\_most\_one(V) \wedge at\_least\_one(V)$$

- Where:

$$at\_least\_one(V) \equiv \bigvee_{v \in V} v$$

$$at\_most\_one(V) \equiv \bigwedge_{1 \le i < |V|} \; \bigwedge_{i+1 \le j \le |V|} \neg(v_i \wedge v_j)$$

CONSTRAINTS

- In each cell there must be a value, $\forall i, j \in \{1,..9\}$:

$$exactly\_one(\{v_{i,j,k} \mid k \in \{1,..9\}\})$$

- Each value used once for each row, $\forall i \in \{1,..9\}$, $\forall k \in \{1,..9\}$:

$$exactly\_one(\{v_{i,j,k} \mid j \in \{1,..9\}\})$$

- Each value used once for each column, $\forall j \in \{1,..9\}$, $\forall k \in \{1,..9\}$:

$$exactly\_one(\{v_{i,j,k} \mid i \in \{1,..9\}\})$$

CONSTRAINTS

- Until now we worked just with at_most_one and at_least_one constraints, for the following instance is important to define also the at_most_k and at_least_k constraints:

$$at\_most\_k(V,k) \equiv \bigwedge_{X \subseteq V} \bigvee_{v \in X} \neg v \qquad |X| = k + 1$$

$$at\_least\_k(V,k) \equiv at\_most\_k(\{\neg v \,|\, v \in V\}, |V| - k)$$

$$exactly\_k(V) \equiv at\_most\_k(V) \wedge at\_least\_k(V)$$

CONSTRAINTS

- Each value must be used exactly once also in each 3 x 3 sub-grid.
- For each $i, j \in \{0,1,2\}, k \in \{1,..9\}$:

$$exactly\_one(\{v_{3i+r,3j+s,k} \,|\, r \in \{1,..3\}), s \in \{1,..3\}\})$$

NURSE SCHEDULING PROBLEM

- In the next example, called nurse scheduling problem, a hospital supervisor needs to create a schedule for n nurses over a fixed day period, subject to the following conditions:
  - Each day is divided into three 8-hour shifts.
  - Every day, each shift is assigned to a single nurse, and no nurse works more than one shift.
  - Each nurse is assigned to a minimum amount of shifts during the given period.

VARIABLES

- We have to represent the shifts assignment for each nurse:

$$S = \{s_{i,j,k} \mid i \in \{1,..,n_{nurses}\}, j \in \{1,..,n_{days}\}, k \in \{1,..,n_{shiftsxday}\}\}$$

- $s_{i,j,k}$ is true iff nurse i work the shift k on day j.

CONSTRAINTS

- In each shift can work just one nurse, so $\forall j \in \{1,..n_{days}\}$, $\forall k \in \{1,..,n_{shiftsxday}\}$:

$$exactly\_one(\{s_{i,j,k} \,|\, i \in \{1,..,n_{nurses}\}\})$$

- Each nurse can work not more than one shift per day, so $\forall i \in \{1,..n_{nurses}\}$, $\forall j \in \{1,..n_{days}\}$:

$$at\_most\_one(s_{i,j,k} \,|\, k \in \{1,..,n_{shiftsxday}\}\})$$

CONSTRAINTS

- If possible, shifts should be distributed evenly and fairly, so that each nurse works the minimum amount of them.

$$min\_shifts\_per\_nurse = \frac{(n_{shifts} * n_{days})}{n_{nurses}}$$

- If this is not possible, because the total number of shifts is not divisible by the number of nurses, some nurses will be assigned one more shift, without crossing the maximum number of shifts which can be worked by each nurse

$$max\_shifts\_per\_nurse = min\_shifts\_per\_nurse + 1$$

CONSTRAINTS

- Finally we add the fair assignment constraints

$$at\_least\_k(\{s_{i,j,k} \mid i \in \{1,..,n_{nurses\}}, j \in \{1,..,n_{days}\}, k \in \{1,..,n_{shiftsxday}\}\}, min\_shifts\_per\_nurse)$$

$$at\_most\_k(\{s_{i,j,k} \mid i \in \{1,..,n_{nurses\}}, j \in \{1,..,n_{days}\}, k \in \{1,..,n_{shiftsxday}\}\}, max\_shifts\_per\_nurse)$$