SAT: Z3 Exercises

Exercises Theory

Angelo Quarta

20 March 2025

Table of Contents

Knights and Knaves

Knights and Knaves

Definition
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 that the inhabitant A says: *Either I am a knave or B 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*, $a_{\text{knight}} = \neg a_{\text{knave}}$.

> 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: $a_{\text{knave}} \vee b_{\text{knight}}$

If a is a_{knight}, then a's statement is true:

$$a_{\text{knight}} \implies (a_{\text{knave}} \vee b_{\text{knight}})$$

If a is a_{knave}, then a's statement is a lie:

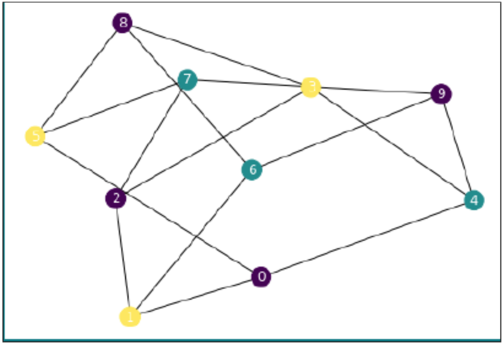
$$a_{\text{knave}} \implies \neg(a_{\text{knave}} \vee b_{\text{knight}})$$

Coloring Graph Problem

Coloring Graph Problem

Definition

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 $n \times d$ boolean variables $v_{i,j}$ with $i \in \{1, \ldots, n\}$, and $j \in \{1, \ldots, d\}$.

$v_{i,j}$ 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 edge must have different colors in its vertices:

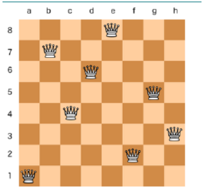
$$\bigwedge_{v_{i,j}\in E} \bigwedge_{1\leq k\leq d} \neg(v_{i,k} \wedge v_{j,k})$$

N-Queens

N-Queens

Definition

Placing n chess queens in a $n \times n$ chessboard so that no queens threatens each other is called the n-queens problem.



Solution

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 $n \times n$ boolean variables $p_{i,j}$ with $i, j \in \{1, \dots, n\}$.

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}

Each variable is true if a queen is in that particular position.

Constraints

At least one queen on each row and column:

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

At most one queen in each row and column:

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

At most one queen in each diagonal:

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

Sudoku

Sudoku

Sudoku is a logic-based, combinatorial number-placement puzzle.

		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		

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.

Variables

> We have to represent the position of the cell in the grid and the number associated to it.

> We need 9×9 variables for the position and other 9 each to fix the number, the total number of variables is $9 \times 9 \times 9$.

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

Constraints

For this instance is important to define the exactly_one constraint, which, given V a set of boolean variables, can be defined as:

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

Where:

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

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

Constraints

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

$$\text{exactly\_one}(\{v_{i,j,k} | k \in \{1, \ldots, 9\}\})$$

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

$$\text{exactly\_one}(\{v_{i,j,k} | j \in \{1, \ldots, 9\}\})$$

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

$$\text{exactly\_one}(\{v_{i,j,k} | i \in \{1, \ldots, 9\}\})$$

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:

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

Therefore,

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

$$\text{exactly\_k}(V) \equiv \text{at\_most\_k}(V) \vee \text{at\_least\_k}(V)$$

Constraints

Each value must be used exactly once also in each 3×3 sub-grid.

For each $i, j \in \{0, 1, 2\}$, $k \in \{1, \dots, 9\}$:

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

Nurse Scheduling Problem

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:

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

$s_{i,j,k}$ is *true* if and only if the nurse i work the shift k on day j.

Constraints

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

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

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

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

Constraints

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

$$\text{min\_shifts\_per\_nurse} = \frac{n_{shifts}\, n_{days}}{n_{nurses}}$$

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.

$$\text{max\_shifts\_per\_nurse} = \text{min\_shifts\_per\_nurse} + 1$$

Constraints

Finally we add the fair assignment constraints:

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

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