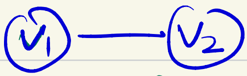
If we just want to check if the given graph is k-colorable for any given value k, we can indeed having "at least one color constraints" suffices.

↳ consider $G(V,E)$; $V = \{v_1, v_2\}$, $E = \{(v_1, v_2)\}$



& $k=2$ {Red, green}

<u>Constraints</u> :-

$$F = \left.\begin{array}{l}(\neg v_1^R \vee \neg v_2^R) \wedge \\ (\neg v_1^G \vee \neg v_2^G) \wedge\end{array}\right\} \text{No adjacent vertex can have same color}$$

$$\left.\begin{array}{l}(v_1^R \vee v_1^G) \wedge \\ (v_2^R \vee v_2^G)\end{array}\right\} \text{A vertex should be assigned at least one color.}$$

Note that with above constraints following assignments are not possiable :

$\sigma - \langle v_1^R \mapsto 1, v_1^G \mapsto 1, v_2^R \mapsto 0, v_2^G \mapsto 0 \rangle$

$\sigma \not\models F$

$\sigma = \langle v_1^R \mapsto 1, v_1^G \mapsto 1, v_2^R \mapsto 1, v_2^G \mapsto 0 \rangle$

$\sigma \not\models F$.

Note that if k is more than the required # of colors then having only "at least one color constraints" can assign multiple colors to a vertex.

Chromatic # :- the minimum # colors necessary for proper coloring of a graph.

Given a graph G(V,G) & k colors; check if k is #chromatic number. If yes, give an proper color assignment for the vertices of graph G.

↓ encode the problem into the problem of satisfiabity.

→ If the corresponding formula F is satisfiable then k is chromatic # for given G(V,G). We can extract the proper coloring assignment from the satisfying assignment.

→ If the corresponding formula is unsatisfiable then k is not the chromatic # for the given graph G(V,E).

Incremental SAT Solving :

starts with k, check if k-colorable if not increase k.

Solving Sudoku

* In a n x n Sudoku, one has to fill a partially filled n x n grid with number 1,.., n such that
  1. each row contains # 1,...,n each number appears exactly once.
  2. each column contains the # 1,...n and each number should appear exactly once.
  3. each of the disjoint $\sqrt{n} \times \sqrt{n}$ (assuming n to be prefect square) sub grids contains the number 1...n.

In a $n \times n$ Sudoku, one has to fill a partially filled $n \times n$ grid with number $1, \ldots, n$ such that

1. each row contains # $1, \ldots, n$ each number appears exactly once.
2. each column contains the # $1, \ldots n$ and each number should appear exactly once.
3. each of the disjoint $\sqrt{n} \times \sqrt{n}$ (assuming n to be prefect square) sub grids contains the number $1 \ldots n$.

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

In a $n \times n$ Sudoku, one has to fill a partially filled $n \times n$ grid with number $1, \dots, n$ such that

1. each row contains # $1, \dots, n$ each number appears exactly once.
2. each column contains the # $1, \dots, n$ and each number should appear exactly once.
3. each of the disjoint $\sqrt{n} \times \sqrt{n}$ (assuming n to be prefect square) sub grids contains the number $1 \dots n$.

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

Sudoku instance $\longrightarrow$ encode $\longrightarrow$ CNF formula F

1. Sudoku is solvable if and only if F is satisfiable & from the satisfying assignment of F, we can decode the solution for sudoku instance.

2. If F is unsatisfiable, then Sudoku problem is not solvable.

1	

2×2

b	c
a	d

2×2

propositional variables $\langle a, b, c, d \rangle$

variable $v \vdash 1$ indicates that at position v, value is 1

variable $v \vdash 0$ indicates that at position v, value is 2

We know that b should be mapped to zero..

b	c
a	d

2×2

propositional variables $\langle a, b, c, d \rangle$

variable $v \vdash 1$ indicates that at position v, value is 1

variable $v \vdash 0$ indicates that at position v, value is 2

Constraints:

* each row should contains number $\langle 1, 2 \rangle$.

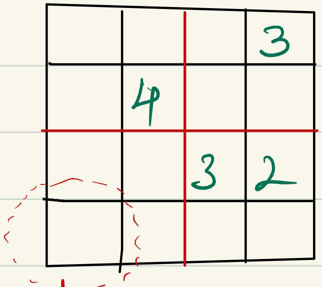
$$b \oplus c$$
$$a \oplus d$$

* each column should contains number $\langle 1, 2 \rangle$

$$b \oplus a$$
$$c \oplus d$$

$$F = (b \oplus c) \wedge (a \oplus d) \wedge (b \oplus a) \wedge (c \oplus d) \wedge (\neg b)$$

Note that F will have unique solution.



we can't have one variable per position.

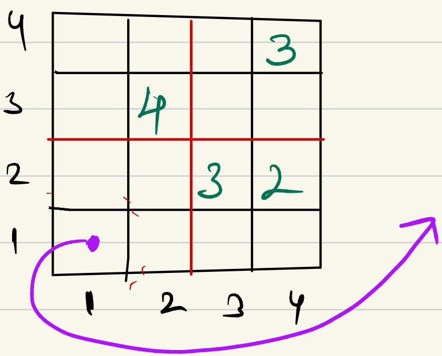
we need to keep track of values also.

↳ let us focus on only this cell, it can take 4 values.

we can have 4 variable for each cell.

n variables

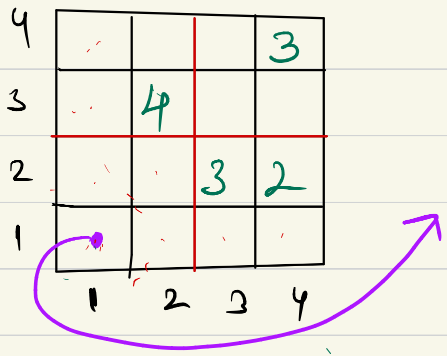
↳ nxn cell



Total n^3 variables :- x_{rcv} : r - row, c - column

v - value.

$x_{111}, x_{112}, x_{113}, x_{14}$



$x_{111}, x_{112}, x_{113}, x_{114}$ — these variables should take **exactly** one value.

→ Repeat for each cell.

Contraint 2: Each row has all the numbers.

first row:

Value 1 should occur at least at one place in a row

$$(x_{111} \vee x_{121} \vee x_{131} \vee x_{141}) \wedge$$

$$(x_{112} \vee x_{122} \vee x_{132} \vee x_{142}) \wedge \cdots \wedge (x_{114} \vee x_{124} \vee x_{134} \vee x_{144})$$

→ Repeat for every row.

4				3
3		4		
2			3	2
1				
	1	2	3	4

Constraint 3: each column has all the numbers.

first column:

$$\underbrace{\begin{array}{l}(x_{111} \vee x_{211} \vee x_{311} \vee x_{411}) \wedge \\ (x_{112} \vee x_{212} \vee x_{312} \vee x_{412}) \wedge \dots \wedge \\ (x_{114} \vee x_{214} \vee x_{314} \vee x_{414})\end{array}}_{\text{Repeat this for every column}}$$

Constraint 4: each block has all the numbers.

first block:

$$(x_{111} \vee x_{121} \vee x_{211} \vee x_{221}) \wedge \dots$$
$$\wedge (x_{114} \vee x_{124} \vee x_{214} \vee x_{224})$$

$$F = \left(\bigwedge_{1 \le r \le n} \bigwedge_{1 \le c \le n} \left(x_{rc1} \vee x_{rc2} \vee x_{rc3} \vee x_{rc4} \right) \right) \wedge$$

$$\left(\bigwedge_{1 \le r \le n} \bigwedge_{1 \le c \le n} \bigwedge_{1 \le v < v' \le n} \left(\neg x_{rcv} \vee \neg x_{rcv'} \right) \right) \wedge$$

each cell has exactly one value

$$\left(\bigwedge_{1 \le r \le n} \bigwedge_{1 \le v \le n} \left(x_{r1v} \vee x_{r2v} \vee x_{r3v} \vee x_{r4v} \right) \right) \wedge$$

each row has all values

$$\left(\bigwedge_{1 \le c \le n} \bigwedge_{1 \le v \le n} \left(x_{1cv} \vee x_{2cv} \vee x_{3cv} \vee x_{4cv} \right) \right) \wedge$$

each column has all values

$$\left(\bigwedge_{1 \le r \le \sqrt{n}} \bigwedge_{1 \le c \le \sqrt{n}} \bigwedge_{1 \le v \le n} \left(\bigvee x_{rcv} \right) \right) \rightarrow$$

each block has all values.

$$F = \Big(\bigwedge_{1\le r\le n} \bigwedge_{1\le c\le n} (x_{rc1} \vee x_{rc2} \vee x_{rc3} \vee x_{rc4}) \Big) \wedge$$

$$\Big(\bigwedge_{1\le r\le n} \bigwedge_{1\le c\le n} \bigwedge_{1\le v<v'\le n} (\neg x_{rcv} \vee \neg x_{rcv'}) \Big) \wedge$$

each cell has exactly one value

$$\Big(\bigwedge_{1\le r\le n} \bigwedge_{1\le v\le n} (x_{r1v} \vee x_{r2v} \vee x_{r3v} \vee x_{r4v}) \Big) \wedge$$

each row has all values

$$\Big(\bigwedge_{1\le c\le n} \bigwedge_{1\le v\le n} (x_{1cv} \vee x_{2cv} \vee x_{3cv} \vee x_{4cv}) \Big) \wedge$$

each column has all values

$$\Big(\bigwedge_{1\le r\le \sqrt{n}} \bigwedge_{1\le c\le \sqrt{n}} \bigwedge_{1\le v\le n} \big(\bigvee x_{rcv} \big) \Big) \cancel{\wedge} \rightarrow$$

each block has all values.

$$x_{324} \wedge x_{233} \wedge x_{242} \wedge x_{324}$$

as per clues

	1	2	3	4
4				3
3		4		
2			3	2
1				

Let us take a detour :-

$$x_1 + x_2 + x_3 + \cdots + x_n \leq 1$$

Pairwise Encoding : $O(n^2)$

↳ Can we do better?

Sequential Encoding :-

Introduced temporary variables $S_1, \ldots S_{n-1}$. such that S_i is assigned true if and only if Sum up to x_i is exactly one.

$$x_1 + x_2 + x_3 \leq 1$$

temp. variable = s_1 & s_2

Come up with the encoding

$$x_1 + x_2 + x_3 \leq 1$$

1. If s_i is set to true then x_{i+1} is set to false as sum is already 1

$$(\neg S_1 \vee \neg x_2) \wedge (\neg S_2 \vee \neg x_3)$$

2. If s_i is set to true then s_{i+1} is also set to true.

$$(\neg S_1 \vee S_2)$$

3. If x_i is set to true then s_i is also true.

$$(\neg x_1 \vee S_1) \wedge (\neg x_2 \vee S_2)$$

$x_1 + x_2 + x_3 \leq 1$

1. If s_i is set to true then x_{i+1} is set to false as sum is already 1

$(\neg s_1 \vee \neg x_2) \wedge (\neg s_2 \vee \neg x_3)$

} $n-1$ clauses.

2. If s_i is set to true then s_{i+1} is also set to true.

$(\neg s_1 \vee s_2)$

} $(n-1)-1$ clauses.

3. If x_i is set to true then s_i is also true.

$(\neg x_1 \vee s_1) \wedge (\neg x_2 \vee s_2)$

} $(n-1)$ clauses

Total: $3n-4$ clauses

at the cost of introducing new $(n-1)$ variables.

Pseudo Boolean Constraints :-

$x_1 \ldots x_n$ Boolean variables.

Pseudo Boolean Constraint:

$$c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \leq c$$

where $c_1, \ldots c_n \in \mathbb{Z}$

$$2x_1 + 3x_2 + x_3 \leq 3$$

Q Solve this using Boolean reasoning.