Sudoku is Hard

Eve Routledge

Complexity Theory

- This puzzle is hard, even for computers to solve.
- We judge difficulty on the time taken for an algorithm to solve a puzzle in relation to it’s input, we can quantify this using big o notation and we classify them into sets
- P are ‘easy’ and can be solved in $O(n^c)$
- NP are ‘difficult’ and cannot be solved in $O(n^c)$
- To show sudoku is hard we must show it belongs to NP, in order to do this we can reduce a known NP problem (in our case SAT) to sudoku in polynomial time.

SAT

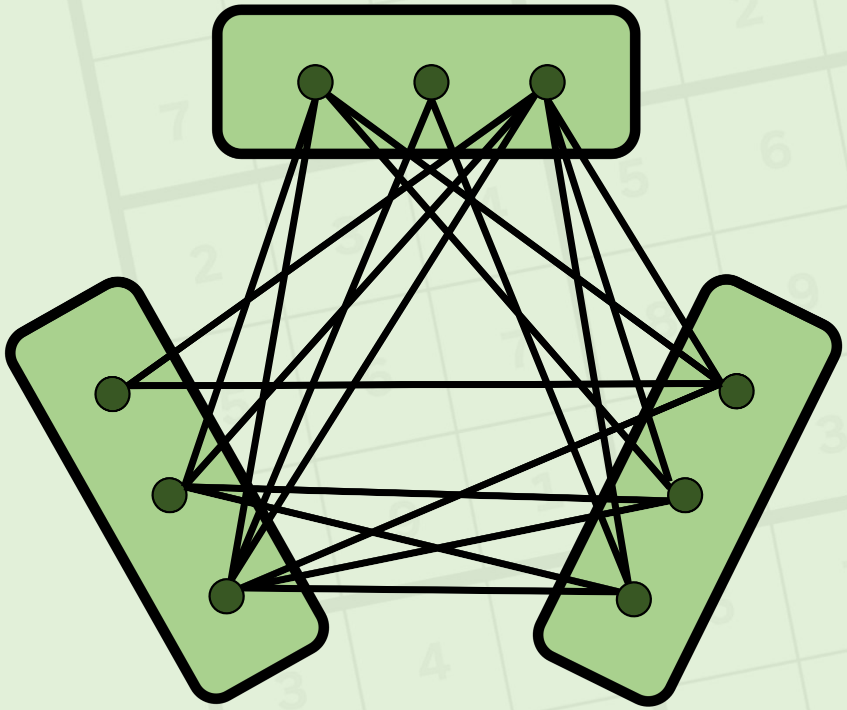
$$(a \lor b) \land (\neg a \lor \neg b)$$

a	b	$(a \lor b)$	$(\neg a \lor \neg b)$	$(a \lor b) \land (\neg a \lor \neg b)$
F	F	F	T	F
F	T	T	T	T
T	F	T	T	T
T	T	T	F	F

Def^n: SAT is the decision problem determining whether a boolean expression is satisfiable or not. Cook Levin Theorem proves this is in NP.

Triangulating a Tripartite Graph

Def^{n}:
A graph G = (V, E) is **Tripartite** if a partition V_1, V_2, V_3 exists such that the vertices are split into three sets with no edges between vertices that belong to the same set, i.e for all $(v_i, v_j) \in E$ if $v_i \in V_i$ then $v_j \notin V_i$.
A **Triangulation** T of a graph G is a way to divide all edges into disjoint subsets T_i , each forming a triangle ($T_i = \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\}$).



Latin Square

Def^n: A valid **Latin Square** puzzle is a function L : i,j → x for values i,j ∈ {1, .., D} and x ∈ {0, ..., D} satisfying the following:

- for all a, b, c ∈ {1, ..., D} with L(a, b) ≠ 0 and L(a, c) ≠ 0 then L(a, b) ≠ L(a, c)
- for all a, b, c ∈ {1, ..., D} with L(a, b) ≠ 0 and L(c, b) ≠ 0 then L(a, b) ≠ L(c, b)

It is complete or solved if for all i,j ∈ {1, ..., D}, L(i, j) ≠ 0

A	B	C
B	C	A
C	A	B

Sudoku

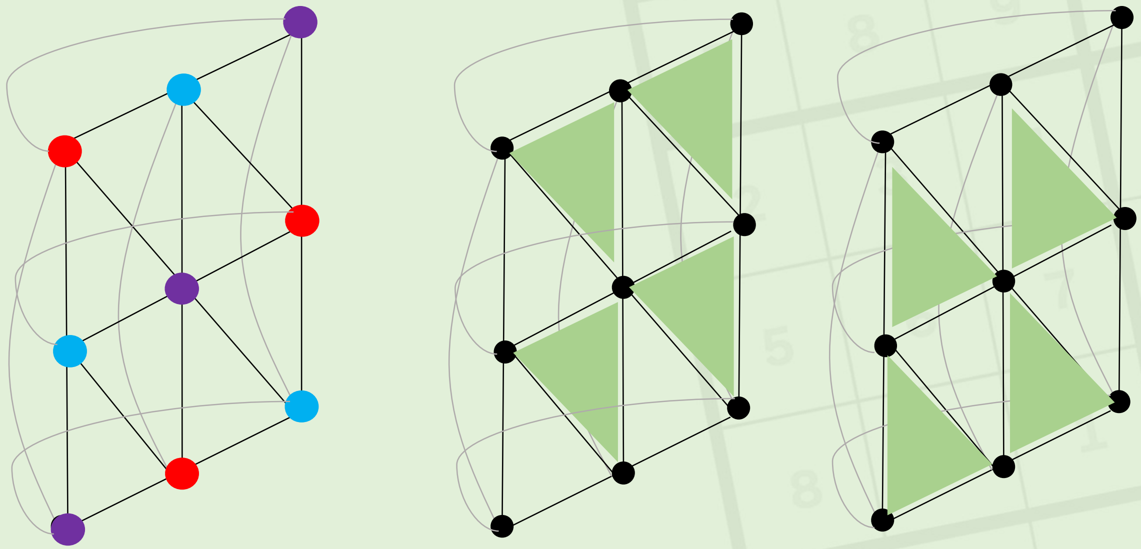
Def^n: A valid **Sudoku** puzzle is a function S : i, j → x for values i, j ∈ {1, ..., D^2} and x ∈ {0, ..., D^2} satisfying the following:

- for all a, b, c ∈ {1, ..., D2} with S(a, b) ≠ 0 and S(a,c) ≠ 0, then S(a, b) ≠ S(a,c)
- for all a, b, c ∈ {1, ..., D^2} with S(a,b) ≠ 0 and S(c,b) ≠ 0, then S(a, b) ≠ S(c,b)
- for all a, b, c, d ∈ {1, ... , D^2} with a mod D = c mod D, b mod D = d mod D, S(a,b) ≠ 0 and S(c,d) ≠ 0, then S(a,b) ≠ S(a,c)

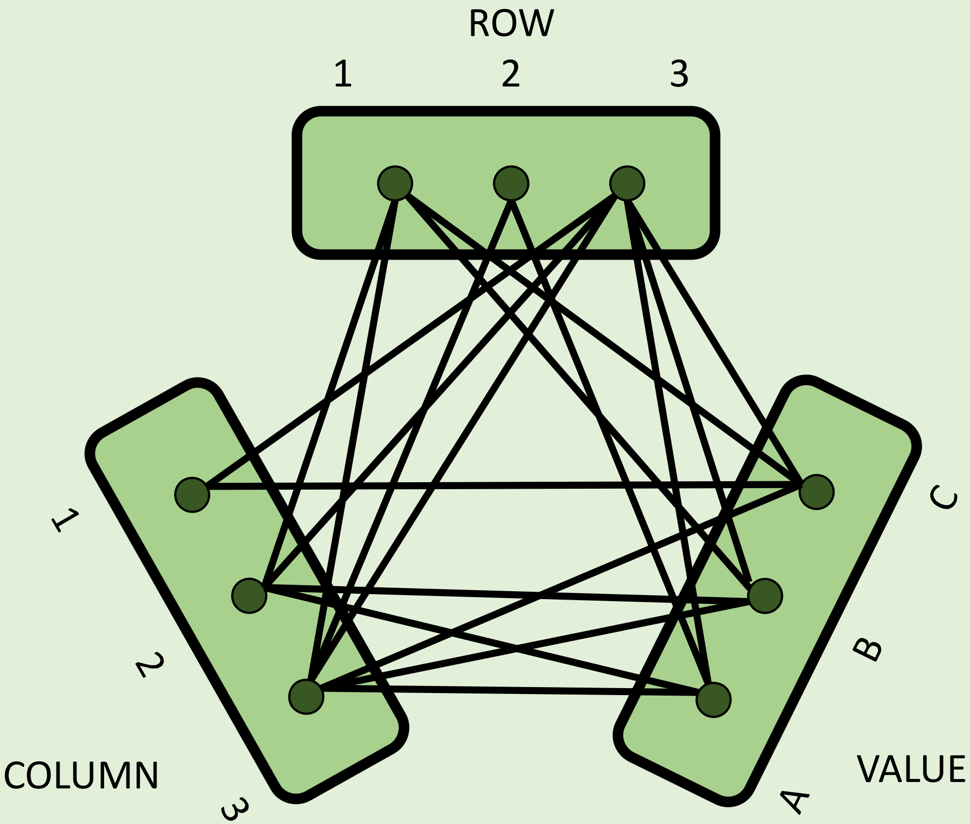
1	2	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	3	4	5	6	7	8	9	1
5	6	7	8	9	1	2	3	4
8	9	1	2	3	4	5	6	7
3	4	5	6	7	8	9	1	2
6	7	8	9	1	2	3	4	5
9	1	2	3	4	5	6	7	8

SAT-> Triangulating a Tripartite Graph

$(a \vee b \vee c) \wedge (\neg a \vee b \vee d)$



Triangulating a Tripartite Graph -> Latin Square



- Each row/column contain each element only once.
- If $(r_i, c_j) \in E$ then LF(i,j)=empty else LF(i,j) $\neq$ k, $k \in [1, \ldots, t]$
- If $(r_i, e_k) \in E$ then for constant i, LF(i,j) $\neq$ k
- If $(c_i, e_k) \in E$ then for constant j, LF(i,j) $\neq$ k

A	B	C
B	C	A
C	A	B

* Framework may only give a r by s array with values 1…t but this can be expanded to n by n dimensions

Latin Square -> Sudoku

Lemma: Let S be a Sudoku problem with
construction

$0\ if$ (i, j) ∈ S_l

S | 1 mod n)n + $\left\lfloor\frac{i-1}{n}\right\rfloor$ + j − 1)

1		
2	**3**	

w)| $\left\lfloor\frac{i-1}{n}\right\rfloor$ = 0 & (j mod n) = 1]

e nentation S' to complete th
puzzle if and only if the square L such tha
(i, j) − 1)/n + 1 for all (i, j) ∈ S_l is a Latin sq

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

Assume we have a quick algorithm for solving sudoku then we can solve SAT with a quick algorithm, contradiction.

Sudoku $\leq_p$Latin Square $\leq_p$ Triangulating a Tripartite Graph $\leq_p$ SAT