Graphsat

~~Recall~~ This asks if we can find a

$a \quad b$
$x \quad y$

Variables
(Domain = $\top/\bot$)

$a \quad \neg a$
$b \quad \neg b$
$x \quad \bar{x}$
$y \quad \bar{y}$

Literals

$a \vee b \vee \neg c$

$\neg x \vee \neg y$

$\neg x$

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

3SAT: $(a \vee b \vee \neg c) \wedge (b \vee \neg c \vee d) \wedge (a \vee b \vee d)$

2SAT: $(a \vee b) \wedge (b \vee \neg c)$

Satisfiable. Else

call it Unsatisfiable.

Facts of life:

① 2SAT $\in P$. If
n ~~[illegible]~~ then we can find a satisfying assignment in $p(n)$ time, where p is some polynomial. And if we know the assignment, then we can ~~check that it is a satisfying assignment~~

② 3SAT $\in NP$. We can check a satisfying assignment in poly. time.

Exponential time.

Bruteforce strategy:

Given x: (Assignment)

such that $\text{assign}(a, x) = T$

③ 3SAT $\notin$ co-NP. If a formula is satisfiable, we can check in poly. time given an assigment. If a formula is unsatisfiable, checking that will take exponential time.

~~Intuition:~~

① Solving analysis assignment: HARD

② Checking

$$① + ② \Rightarrow \in NP$$

③ Checking assignment if solution is incorrect: HARD

$$① + ③ \Rightarrow \notin \text{co-NP}$$

$2SAT \in P$

Not sure why this boundary exists!

$3SAT \in NP$

$$SAT(x) := \exists\, a,\ assign(a, x) = T$$

~~GraphSAT(~~

$a \leftrightsquigarrow$

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

$$(a \vee b) \wedge (b \vee c) \wedge (a \vee c) \wedge (c \vee \bar{d})$$

$(a \vee$

4^5 formulas.

$$\vdots$$

$$(\bar{a} \vee \bar{b}) \wedge (\bar{b} \vee \bar{c}) \wedge (\bar{a} \vee \bar{c}) \wedge (\bar{c} \vee \bar{d})$$

A graph g is SATISFIABLE if **every** formula $x \in g$ is SATISFIABLE.

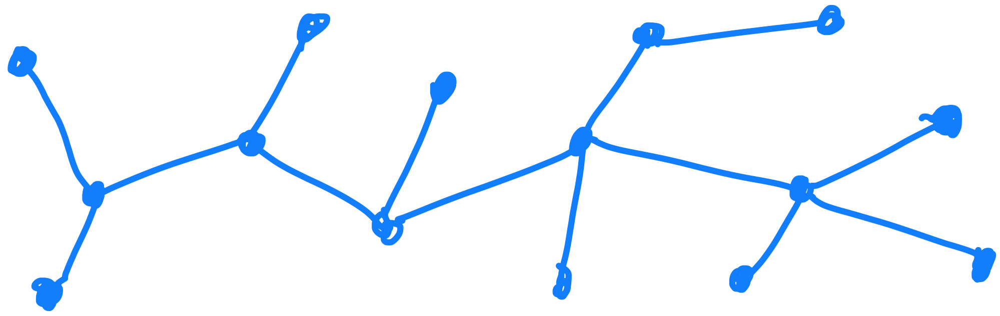
A graph g is UNSATISFIABLE if **any** formula $x \in g$ is .

Q0. Which graphs are satisfiable/unsatisfiable?

Q1. How much more complicated is GraphSAT compared to SAT.

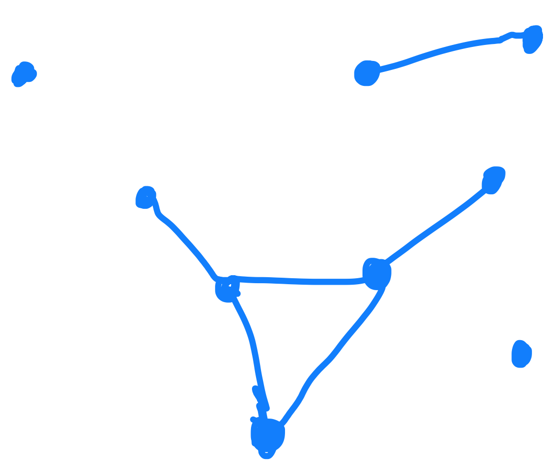
I will state the answers to Q0 as **facts** instead of results. → More on Friday @ 4 pm CST

- Any finite tree is SATISFIABLE



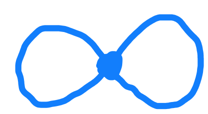
- Any cycle graph is SATISFIABLE

- [redacted]



This whole thing is SATISFIABLE.

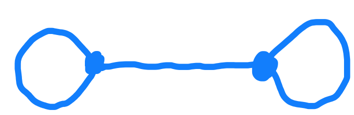
Graphs known to be UNSAT:



$a \wedge \neg a$

Double loop

"smooth butterfly"



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

"Smooth bowtie"

Dumbbell

"smooth $K_{1,1,3}$"

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

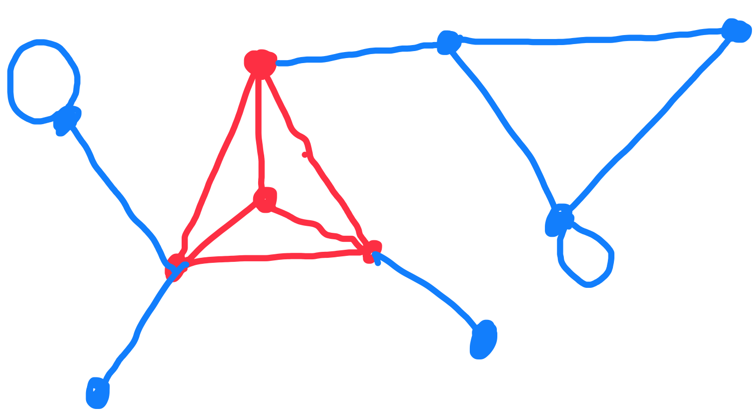
Structure theorem #0

~~SATIS~~

Structure theorem #1 :

Let g be ~~g~~

g is UNSATISFIABLE $\Rightarrow$ h is UNSATISF.



~~is unsatis~~

Structure theorem #2:

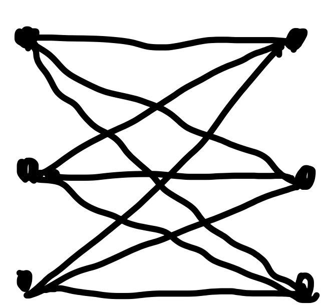
All paths can be smoothed without changing SATISFIABILITY.

Algorithm idea: Check for UNSATISFIABLE

"contain"

~~these should be subgraphs of g.~~

Cannot obtain K_5 or $K_{3,3}$ by smoothing paths in g.



$K_{3,3}$

Q. How quickly can we decide if an arbitrary graph ~~g~~ (ala Kuratowski) graph h?

~~GRAPH MINOR~~

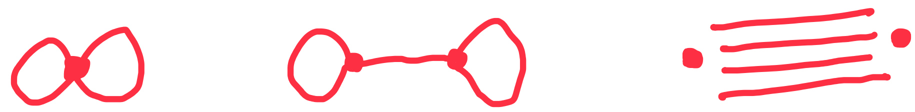
Q. How quickly can we decide if a fixed graph g is "contained" in an arbitrary graph h?

$\in P$

This means finiteness of Kuratowski's

Our result:

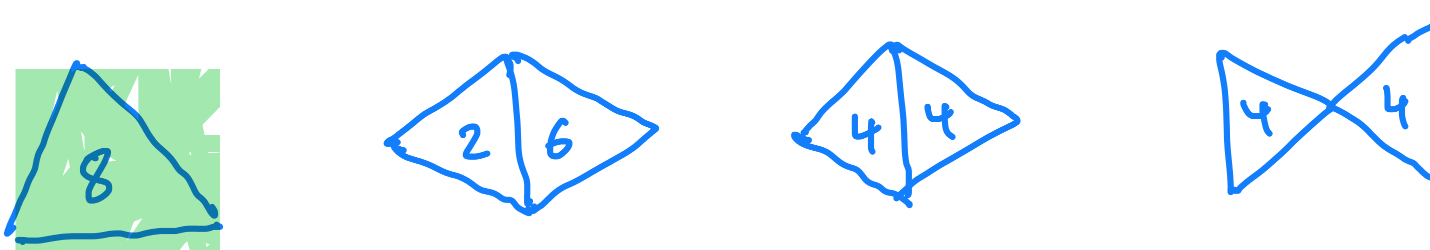
iff it does not "contain" → more relaxed

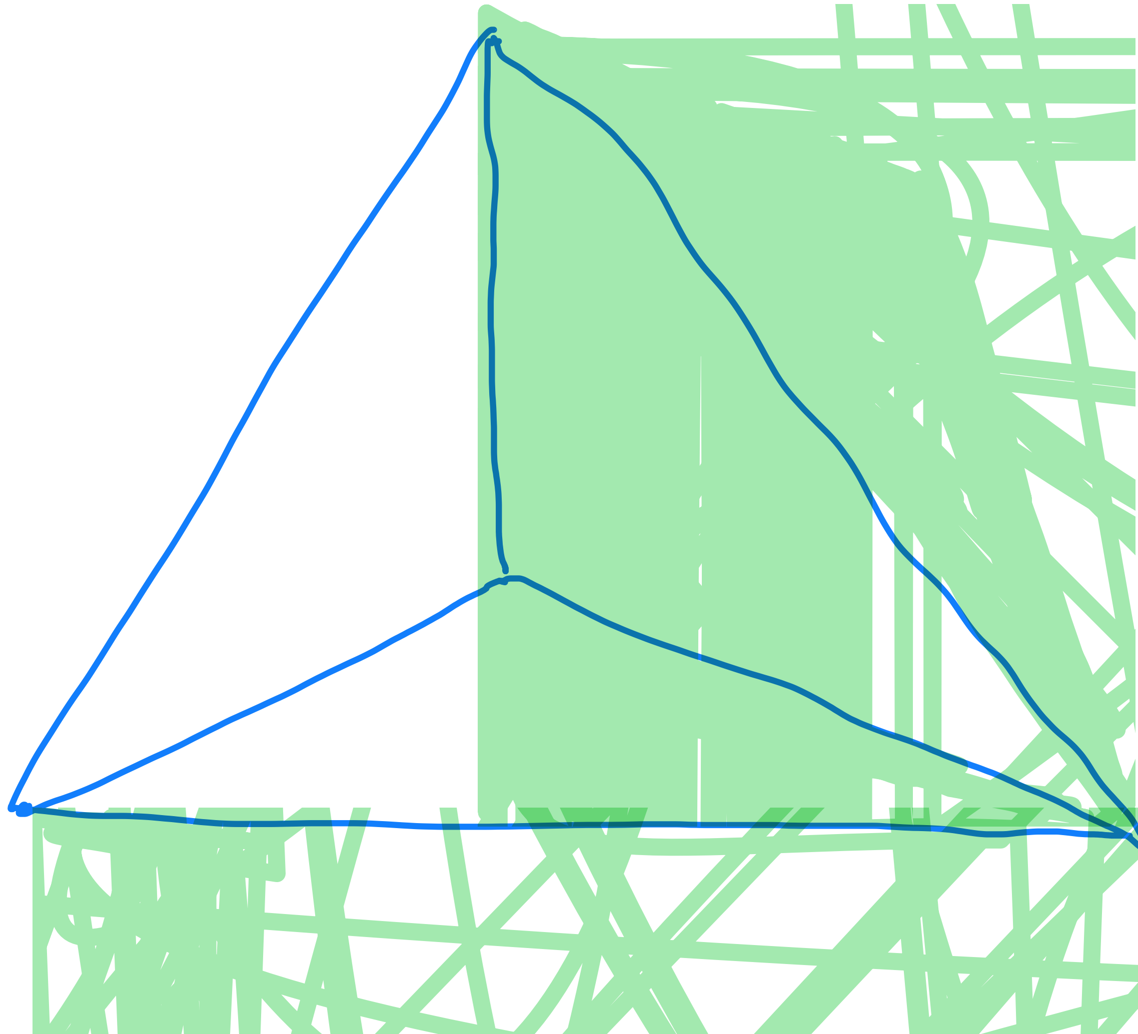


New problem:

The above result only holds for simple graphs (edge size ≤ 2).

For hypergraphs, we don't finite list.





2SAT $\in$ P and 2GraphSAT $\in$ P

3SAT $\in$ NP-complete. 3GraphSAT $\in$???

3GraphSAT (bruteforce) $\notin$ NP

- Given an arbitrary graph that is SATISFIABLE.
- To verify that it is SATISFIABLE, we need to check that graph are SATISFIABLE.
- Each check can be done in P time $\because$ 3SAT $\in$ NP.

than P time.

- 3GraphSAT $\notin$ NP

-

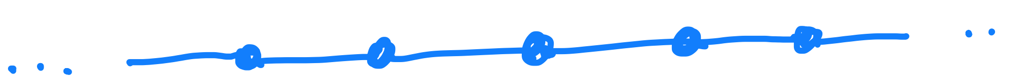
unsatisfiability we need to check

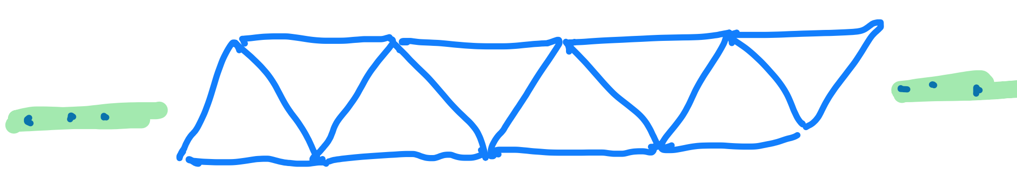
- This boils down to checking a single.
- But 3SAT $\notin$ ~~co-NP $\therefore$ 3Gra~~

Bonus: Infinite graphs

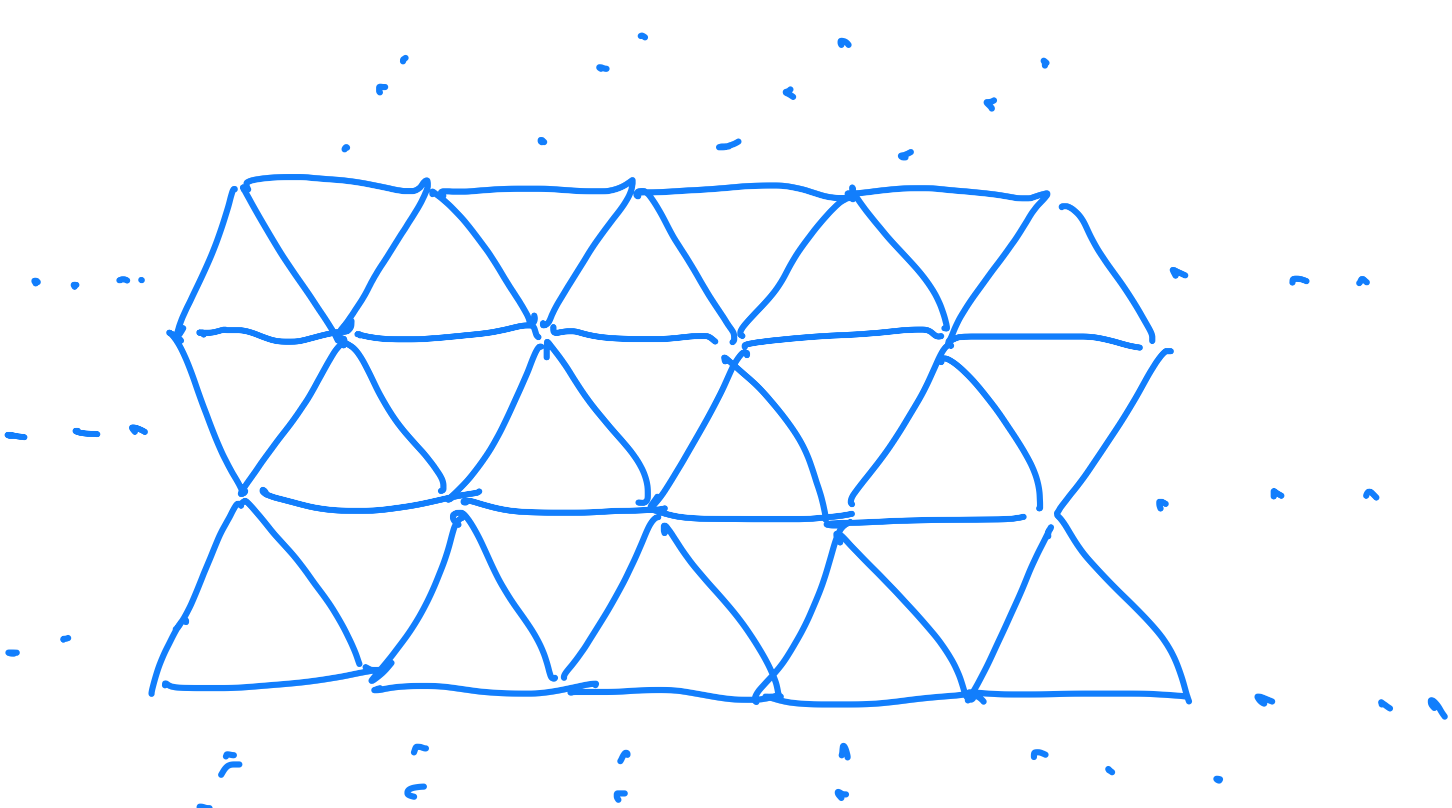
Graphsat can be extended to infinite graphs!

about complexity classes for infinite inputs.

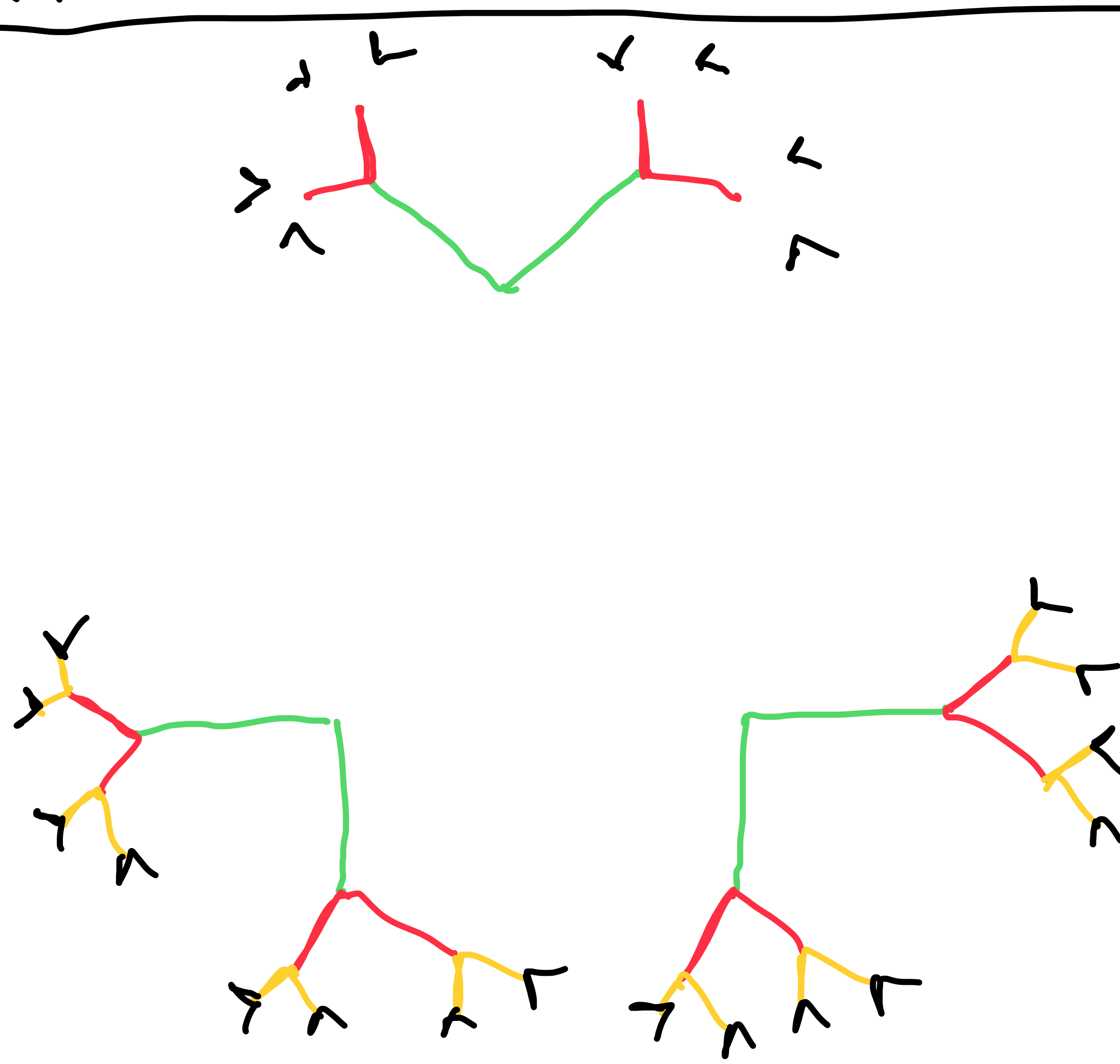




infinite strip of triangles



Known UNSATISFIABLE infinite graph:



Infinite tree graph with uniform degree 3.