P (stands for "Polynomial")

The set of all decision problems that have an algorithm that runs in time $O(n^k)$ for some constant k (on input of size n).

NP (stands for "nondeterministic polynomial")

The set of all decision problems such that for every YES-instance (of size n), there is a certificate (of size $O(n^k)$) for that instance which can be verified in polynomial time.

NP-hard

The problem B is NP-hard if
for all problems A in NP, A reduces to B.

NP-Complete

The problem B is NP-complete if B is in NP
and B is NP-hard

If we had a nondeterministic computer…

Can you think of a polynomial time algorithm on a nondeterministic computer to:

Solve 2-COLOR?

Solve 3-COLOR?

What's 3-SAT?

"AND" of "ORs"
$\wedge$ outside parens
$\vee$ inside parens

Input: A list of Boolean variables $x_1, \dots, x_n$

An expression in Conjunctive Normal Form, where each clause has exactly 3 literals.

One of the subexpressions inside parens

Something like:

$$(z_i \vee z_j \vee z_k) \wedge (z_i \vee z_\ell \vee z_a) \wedge \cdots \wedge (z_a \vee z_b \vee z_c)$$

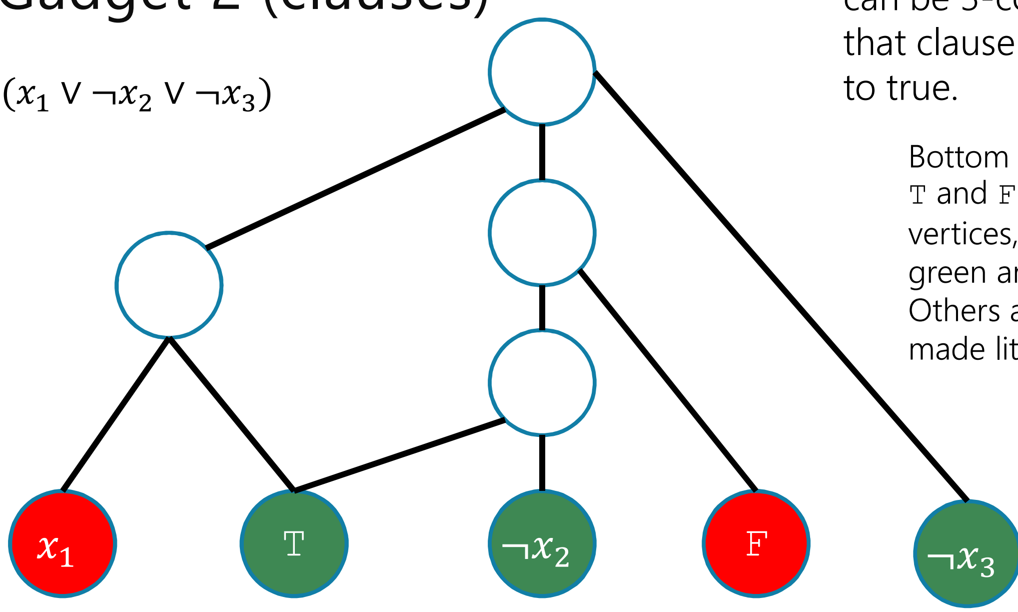
Where z is a "literal" a variable or the negation of a variable ($x_i, \neg x_j$, etc.).

Output: true if there is a setting of the variables where the expression evaluates to true, false otherwise.

Why is it called 3-SAT? 3 because you have 3 literals per clause
SAT is short for "satisfiability" can you satisfy all of the constraints?

Gadget 2 (clauses)

$(x_1 \vee \neg x_2 \vee \neg x_3)$



This tricky little graph can be 3-colored iff that clause evaluates to true.

Bottom row:
`T` and `F` are new vertices, colored green and red.
Others are already-made literal vertices