Recall that we've seen many NP-completeness proofs.

Today – finish the topic of NP-completeness

Circuit-SAT $\leq_p$ 3-SAT $\leq_p$ IND.SET $\leq_p$ VERTEX COVER $\leq_p$ SET COVER

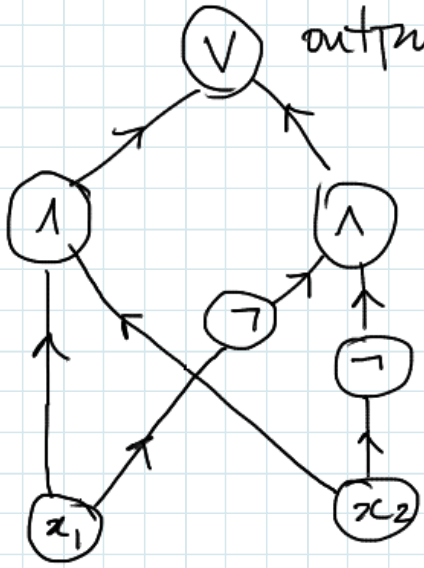
3-SAT $\leq_p$ HAM. CYCLE $\leq_p$ TSP

3-SAT $\leq_p$ SUBSET SUM

today

Last day we saw why Circuit-SAT is NP-complete (at least the outline)

Recall Circuit Satisfiability



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

i.e. x_1 same as x_2

$x_1 \equiv x_2$

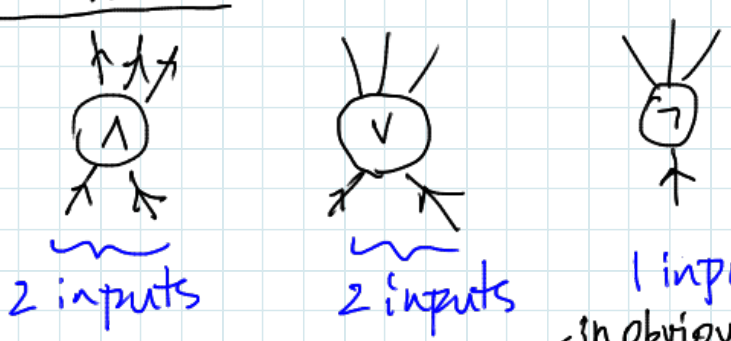
A <u>circuit</u> is a directed acyclic graph

<u>sources</u> (no edge entering) are labelled with variables or 0 or 1 — these are inputs

<u>sink</u> (no edge leaving)

there is one sink — output

<u>internal nodes</u>



A circuit <u>computes an output</u> (in obvious way) when values are given to input variables

e.g. above $x_1 = 0$ $x_2 = 1$ outputs 0

(compute values at internal nodes from sources to sink)

Recall:

Circuit Satisfiability

Input: A circuit C.

Q: Is there an assignment of values to inputs s.t. output is 1? (Is C satisfiable?)

Thm [pf. outline last day] Circuit Sat. is NP-complete

New result for today.

Thm 3-Satisfiability is NP-complete.

Pf. ① $\in$ NP

② Circuit Sat $\leq_p$ 3-SAT

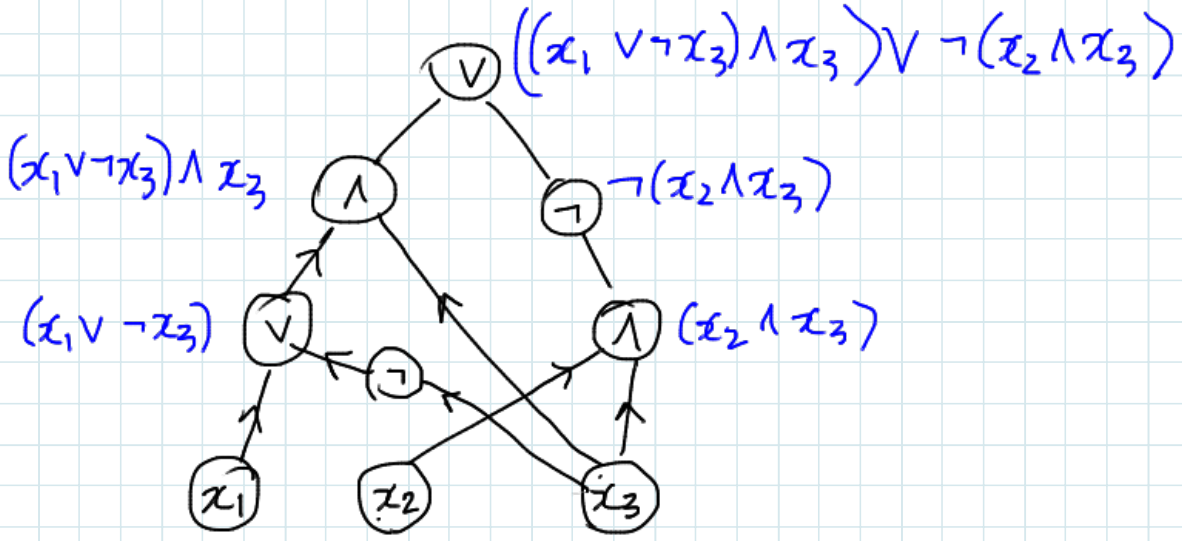
Intuitively, circuits and formulas are equivalent.

Assume a poly. time alg. for 3-SAT

Give a poly. time alg. for circuit Sat.

Input is a circuit C. Create formula F s.t.

C is satisfiable iff F is satisfiable



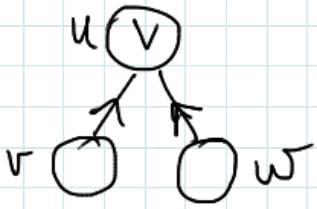
The obvious way to make a formula

Caution Is this poly. size?

No, because the size of the formulas doubles as we go up each level

Better approach:

Make a variable for each node u in circuit

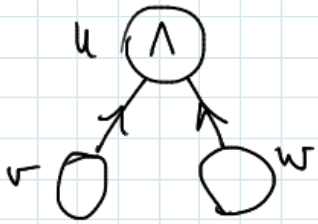


$x_u \equiv x_v \vee x_w$

as clauses

$(\neg x_u \vee x_v \vee x_w) \wedge (x_u \vee \neg x_v) \wedge (x_u \vee \neg x_w)$

$x_u \Rightarrow x_v \vee x_w$ $\quad$ $\neg x_u \Rightarrow \neg x_v \wedge \neg x_w$



$x_u \equiv x_v \wedge x_w$

$(x_u \vee \neg x_v \vee \neg x_w) \wedge (\neg x_u \vee x_v) \wedge (\neg x_u \vee x_w)$

$x_u \Rightarrow x_v \wedge x_w$

$\neg x_u \Rightarrow \neg x_v \vee \neg x_w$

$x_u \equiv \neg x_v$ $\quad$ one is T, one is F

$(x_u \vee x_v) \wedge (\neg x_u \vee \neg x_v)$

Note: $a \equiv b$ is equiv. to $(\neg a \vee b) \wedge (a \vee \neg b)$

Claim We can turn clauses of size 2 into clauses of size 3 with an extra variable x_{new}

$(a \vee b) \longrightarrow (a \vee b \vee x_{new}) \wedge (a \vee b \vee \neg x_{new})$

Take $\wedge$ of all clauses, $\wedge$ x_{output}. Call this F

output = 1

EX. Carry this out on an example

Claim 1 F has poly. size and can be computed in poly. time.
Claim 2 F is satisfiable iff C is satisfiable

Pf

$\Rightarrow$ Suppose F is satisfiable. Then there is an assignment of True/False to the variables
(original inputs + variables assigned to circuit nodes)
that makes F True.
Then x_{output} = True
And, by construction, the variables attached to circuit nodes capture the circuit computation.
Therefore C is satisfiable.

$\Leftarrow$ Suppose C is satisfiable.
Then assigning True/False to variables of F corresponding to C's computation will satisfy F.

□

X

Major open question $P \stackrel{?}{=} NP$

if $P = NP$
all the NP-complete problems can be done in poly. time

if $P \neq NP$
none of the NP-complete problems can be done in poly. time

Thm if $P \neq NP$ then there are problems here.

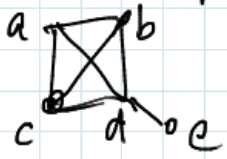
Almost all "natural" problems in NP are known to be in P or NP-complete.

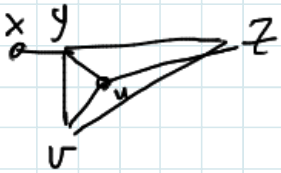
famous book on NP-completeness

Exceptions Open in Garey & Johnston '79

Linear Programming	in P in 1980
Primality	in P in 2002
Graph Isomorphism	still open

given two graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ are they the same graph up to relabelling?

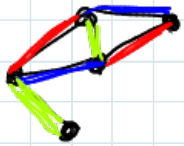




Some interesting NP-complete problems

1. Edge colouring

   Colour edges of a graph s.t. incident edges have different colours



degree $d \Rightarrow$ need d colours

Δ = max degree $\Rightarrow$ need Δ colours

Thm [Vizing] min. no. colours is Δ or $\Delta+1$

But deciding if a graph can be coloured w/ Δ colours is NP-complete.

2. Erik Demaine et al. 2012

   Nintendo games are NP-hard

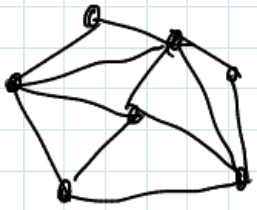
   http://arxiv.org/pdf/1203.1895v1.pdf

   – reductions from 3-SAT.

3. Minimum weight triangulation — NP-hard.

   Mulzer & Rote J. ACM 2008

   http://dl.acm.org/citation.cfm?id=1346336



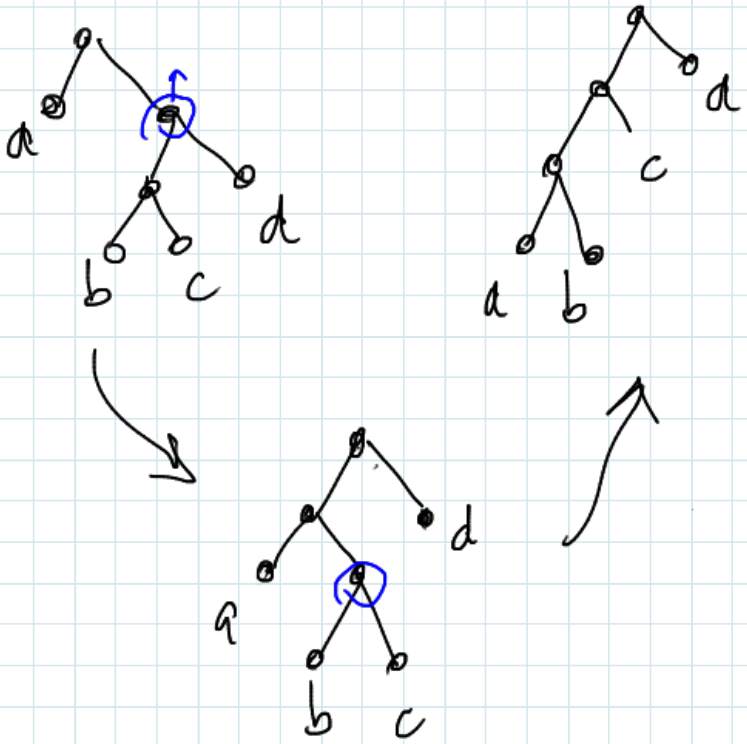
Given pts in plane and $k \in \mathbb{N}$ is there a triangulation with sum of edge lengths $\leq k$

Open Problems:

Are the following problems NP-complete?

1. Given two binary search trees on leaves $1 \ldots n$, and given $k \in \mathbb{N}$, can we get from one to the other using $\leq k$ rotations?

e.g.



2. Given $n \times n \times n$ Rubik's cube (messed up), can we solve it in $\leq k$ moves?

"Move" = rotating a slice

Note: for $3 \times 3 \times 3$ the question of poly. time alg. is moot because there's a finite set of possibilities

proved NP-complete 2017