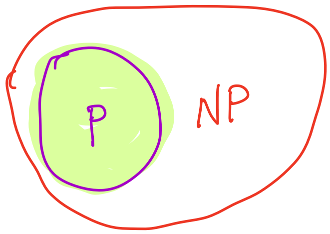
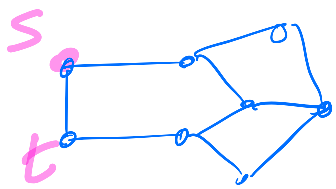
Chapter 7.4 NP-completeness.

As discussed last class, ∃ two categories of complexity that capture most of the problems we are interested in finding algorithms for:



P = languages we can decide in poly-time

NP = languages we can decide in non-det poly time
i.e. we could verify membership if given an appropriate certificate, in det poly-time



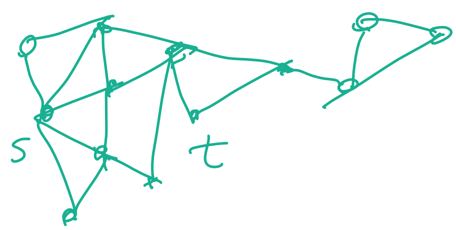
Eg. HamPath ∈ NP because ∃ a certificate (the hamilton path) that we can check in poly time to verify the graph has a hamilton path.

Eg. CLIQUE ∈ NP because "a clique of size k" is a certificate - can check in poly-time that the vertices in the proposed clique are indeed all adjacent to one another.

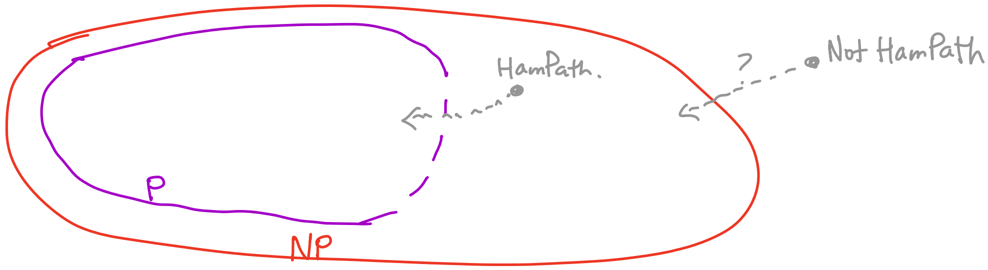
Eg. NotHamPath = $\{\langle G, s, t\rangle \mid \nexists$ ham path from s to t$\}$

Is this language ∈ NP?

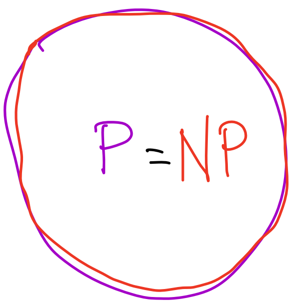
$\overset{?}{\notin}$ NP . . .



April 8, 2025



could be



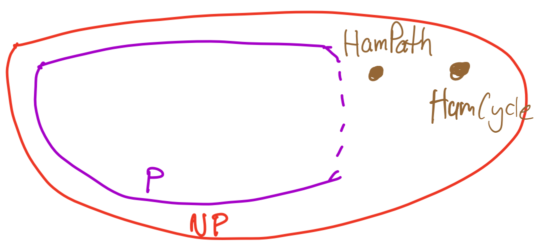
Most computer scientists believe $P \neq NP$, but so far we have not been able to prove that there are some problems are (exponentially) hard to solve but (polynomially) easy to check.

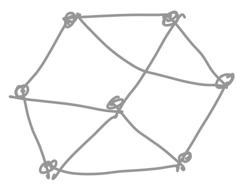
(decide) (prove $\in L$)

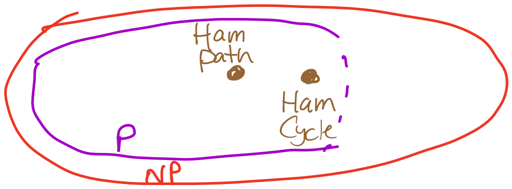
(verify) (confirm that the proof is valid.)

However....

We have a system for identifying a certain relationship among languages $\in NP$...







$HC \leq_p HP$

$HC =$ "on input $\langle G \rangle$

⋮

$HP(\langle G', s, t \rangle)$

⋮

$HP(\langle G', s', t' \rangle)$

⋮

if ___, ACCEPT

else REJECT."

Claim

If $\exists$ a poly time HamPath decider,

Then $\exists$ a poly time HamCycle decider.

Proof

S'pose $\exists$ a poly time HamPath decider $\boxed{HP}$

Then we can construct a poly time HamCycle decider $\boxed{HC}$ as follows:

$\boxed{HC}$ = "on input $\langle G \rangle$, where G is a graph:

1. $\forall$ edge $(u,v) \in V(G)$,

   if $\boxed{HP}(\langle G,u,v \rangle) = \text{accept}$, ACCEPT.

2. If no such edge led to acceptance, REJECT."

The above reduction from HamCycle to HamPath is poly-time...

ie if HP is poly-time, so is HC. If HC is not $\in P$, Neither is HP.

Analyze the work done in HC:

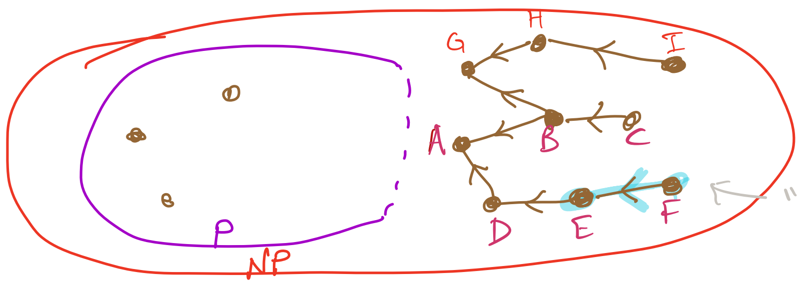
$|E(G)| \leq n^2$

$\therefore$ HC consists of $\leq n^2$ calls to HP.

$\therefore$ if HP is polynomial, so is HC. ∎

We write $HC \leq_p \boxed{HP}$ ← "black box" subroutine

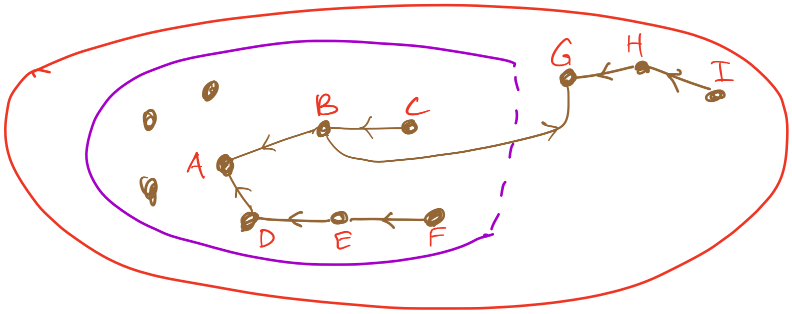
"reduces in poly-time to"



What does poly-time reducibility mean?

"$F \le_p E$"

if we drag A into P...



... every problem reducible to A goes with it into P.

$\in P$

$\in NP$

Boolean Formulas in DNF (Disjunctive Normal Form).

$$(x \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge w \wedge z \wedge \bar{z}) \vee (x \wedge y \wedge \bar{w})$$

Is it satisfiable? $\boxed{x = T,\ y = T,\ w = F}$

ie $\exists$ an assignment of truth values to the variables so that the Boolean Formula evaluates to TRUE?

Yes, in that case ... and in fact DNF formulae can be checked for satisfiability in linear time.

Boolean Formulas in CNF (Conjunctive Normal Form)

$(x \vee \bar{y} \vee w) \wedge (\bar{x} \vee y \vee \bar{w} \vee z) \wedge (x \vee y)$

Is it satisfiable? $x = T$ $y = T$ $w = T$, $z = T$.

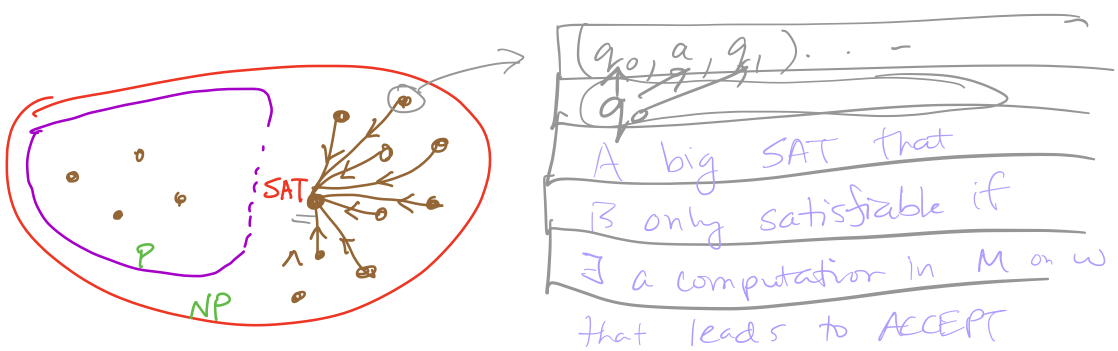
How about

$(x \vee y \vee w) \wedge (\bar{x} \vee \bar{y} \vee \bar{w}) \wedge (x \vee y \vee \bar{w}) \wedge (\bar{x} \vee \bar{y} \vee w) \wedge (x \vee \bar{y} \vee \bar{w})$
$\wedge (\bar{x} \vee \bar{y} \vee \bar{w})$

SAT $= \{ \langle \phi \rangle \mid \phi$ is a Boolean Formula in $\boxed{\text{CNF}}$, and
ϕ has a satisfying assignment$\}$.

assign $\{T, F\}$ to each variable, ϕ = true

Big Theorem [Cook; Levin] $\forall X \subseteq P$, $X \leq_p$ SAT



Defn We call a language Z NP-complete (NP-c) if
$\forall X \in P$, $X \leq_p Z$.   Eg SAT

$3SAT = \{ \langle \phi \rangle \mid \phi$ is a Boolean Formula in 3-CNF, and $\exists$ an assignment to ϕ's variables that makes it TRUE $\}$.

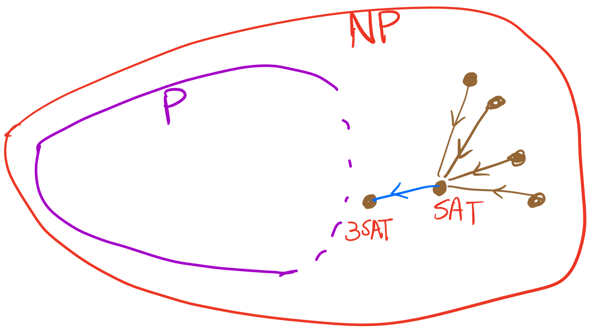
$$\underbrace{(x \vee y \vee z)}_{\text{"OR" clause}} \wedge (\bar{x} \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z})$$

has $\underline{\underline{3}}$ literals

Theorem: $\underline{SAT} \leq_p 3SAT$: Given a poly-time 3SAT decider, S3, we can construct a poly-time SAT decider S

SAT: Known hard. NP-c

3SAT: candidate for NP-c



S = "on input $\langle \phi \rangle$, a Boolean Form in CNF.

1. construct an instance ϕ' of 3SAT, according to an alg.
2. Run S3 on ϕ'
   - if S3 accepts, ACCEPT
   - if S3 reject, REJECT."

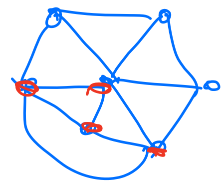
Alg: 1. $(x_1 \vee x_2) \Rightarrow (x_1 \vee x_2 \vee x_2)$

2. $(x_1 \vee x_2 \vee x_3 \vee x_4) \in \underset{SAT}{\phi}$

$\Rightarrow (x_1 \vee x_2 \vee A) \wedge (\bar{A} \vee x_3 \vee x_4) \in \underset{3SAT}{\phi'}$

$(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5) \Rightarrow (x_1 \vee x_2 \vee A) \wedge (\bar{A} \vee x_3 \vee B) \wedge (\bar{B} \vee x_4 \vee x_5)$, etc.

Theorem: 3SAT $\leq_p$ CLIQUE.



Proof: (Sketch - for a particular example)

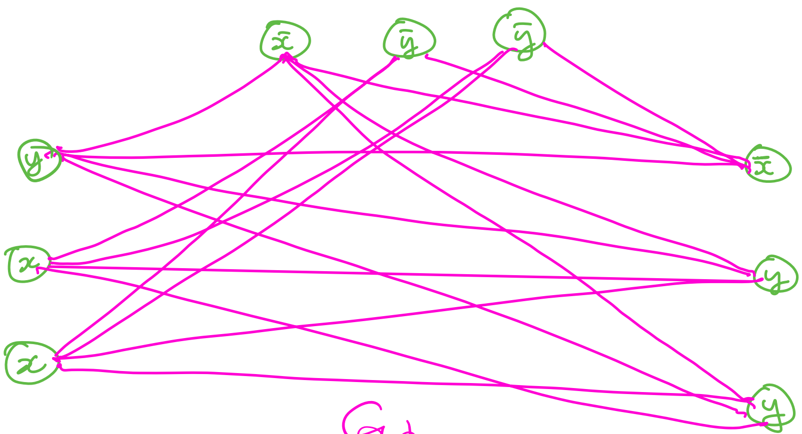
$\phi = (x \vee x \vee y) \wedge (\bar{x} \vee \bar{y} \vee \bar{y}) \wedge (\bar{x} \vee y \vee y)$

c_1 c_2 $c_3 = c_K$

$\approx 3K$

$Y = T$

$X = F$



G_ϕ

The selected 3-clique must have one vertex in each clause, and no two of them can contradict ie can't pick X and $\bar{X}$.

If the graph has a clique of size C (= # clauses in ϕ)

then that identifies an assignment of truth values to the variables -

$\Rightarrow$ the clique contains one literal from each clause.

$\Rightarrow$ there are no contradictions

$\Rightarrow$ ϕ is satisfiable

If $\exists$ a satisfying assignment of $\{T, F\}$ to the variables...

then for each clause, a literal is made true

- pick that literal in the graph, for each clause
- they form a clique in the graph, and there are c of them

Thus $\exists$ a clique of size c $\Longleftrightarrow$ ϕ is satisfiable $\blacksquare$

$\therefore$ 3SAT $\leq_p$ Clique. $\therefore$ CLIQUE $\in$ NP-c.

NP-c

We also need:

we can construct the graph (the input for the CLIQUE-decider)

in poly-time from ϕ.

$$f: \phi \rightarrow (G, K)$$

where f is poly-time computable.

Coles Notes:

- we want poly-time algs for problems (languages)
- if we can't find one... perhaps NONE CAN EXIST
- ... we don't know how to prove that,
  BUT maybe we can prove it is NP-complete.
- If it is NP-c then a poly-time solution exist only if P=NP.
- To show it is NP-c, we reduce a hard problem to ours.

eg Is X hard?

Reduce HamPath to Ham Cycle.

Suppose we have a HamCycle decider HC.

UHP = " on input ⟨G, s, t⟩ where G is a graph,

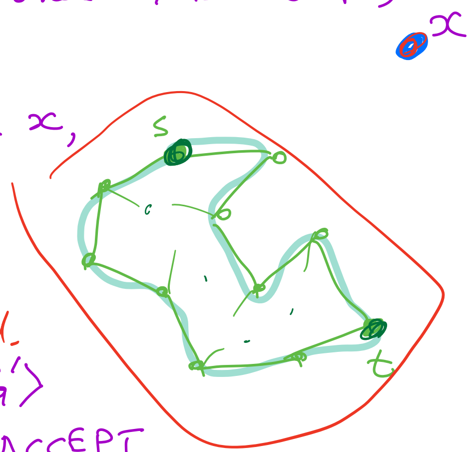
undirected.

$s, t \in V(G)$:

1. Add a new vertex x, with edges to s and t; call result G′
2. Run UHC on ⟨G′⟩

undirected.

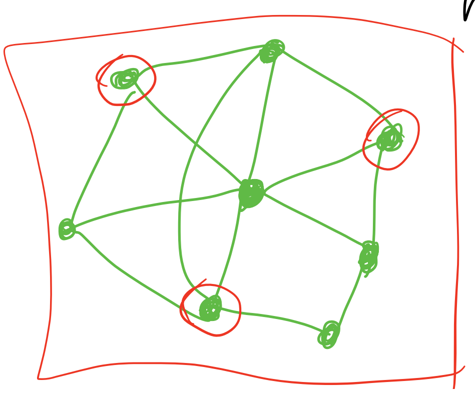
3. If UHC accepts, ACCEPT.
If UHC rejects, REJECT. "



Claim: ∃ HamCycle in $G + (x, s) + (t, x)$ ⟺ ∃ an s–t HamPath in G.

Proof: ... prove this, and then the reduction is complete.

IndepSet = { ⟨G, K⟩ | G is a graph and G has K vertices that are all non-adjacent to one another }



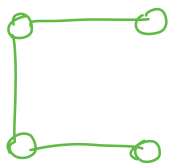
IndepSet ∈ NP ? ✓

IndepSet ∈ NP-c ?

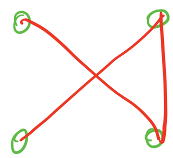
⟨G, K, $v_1, v_2, v_3 .. , v_K$⟩

$$\bar{G} = \begin{cases} \text{- same Vertex Set} \\ \text{- } (i,j) \in E(G) \text{ iff } (i,j) \notin E(\bar{G}). \end{cases}$$

eg.



G



$\bar{G}$

Clique $\le_p$ Indep Set

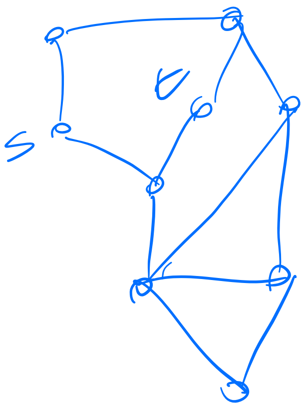
X = "Given $\langle G, k \rangle$,

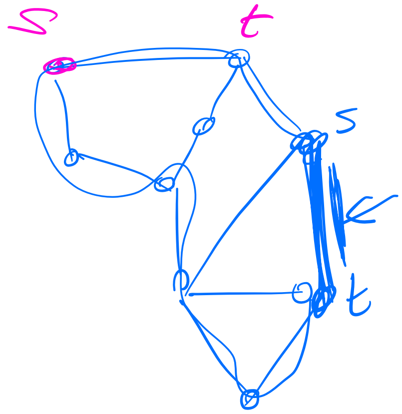
- construct $\bar{G}$
- run $\boxed{\text{IndepSet}(\langle \bar{G}, k \rangle)}$
- if accepts, ACCEPT,
  if rejects, REJECT."

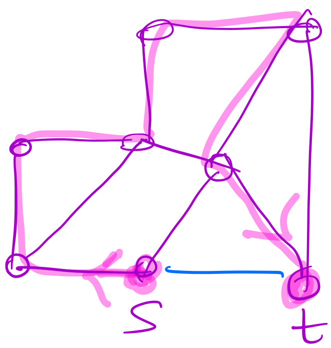
X decides Clique in Poly time.

have HamPath : $G, s, t, s \neq t$.

Ham Cycle G.







$$\phi = (x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4 \vee x_5) \wedge (x_1)$$

$$\phi' = (x_1 \vee \bar{x}_2 \vee A) \wedge (\bar{A} \vee x_3 \vee B) \wedge (\bar{B} \vee x_4 \vee x_5)$$

$$\wedge \; (x_1 \vee x_1 \vee x_1)$$