Some polynomial reductions.

Theorem (Cook–Levin): CNF-SAT is NP-complete.

Proof: (Take CSCE 551)

Prop: CNF-SAT $\leq_p$ IS (Indep. Set)

Cor: IS is NP-complete.

Proof of the Prop: Given an arbitrary cnf formula

$$\varphi = C_1 \wedge \cdots \wedge C_k$$

as input, we construct (in p-time) a graph G and number k' such that φ is satisfiable iff G has an i.s. of size $\geq k'$:

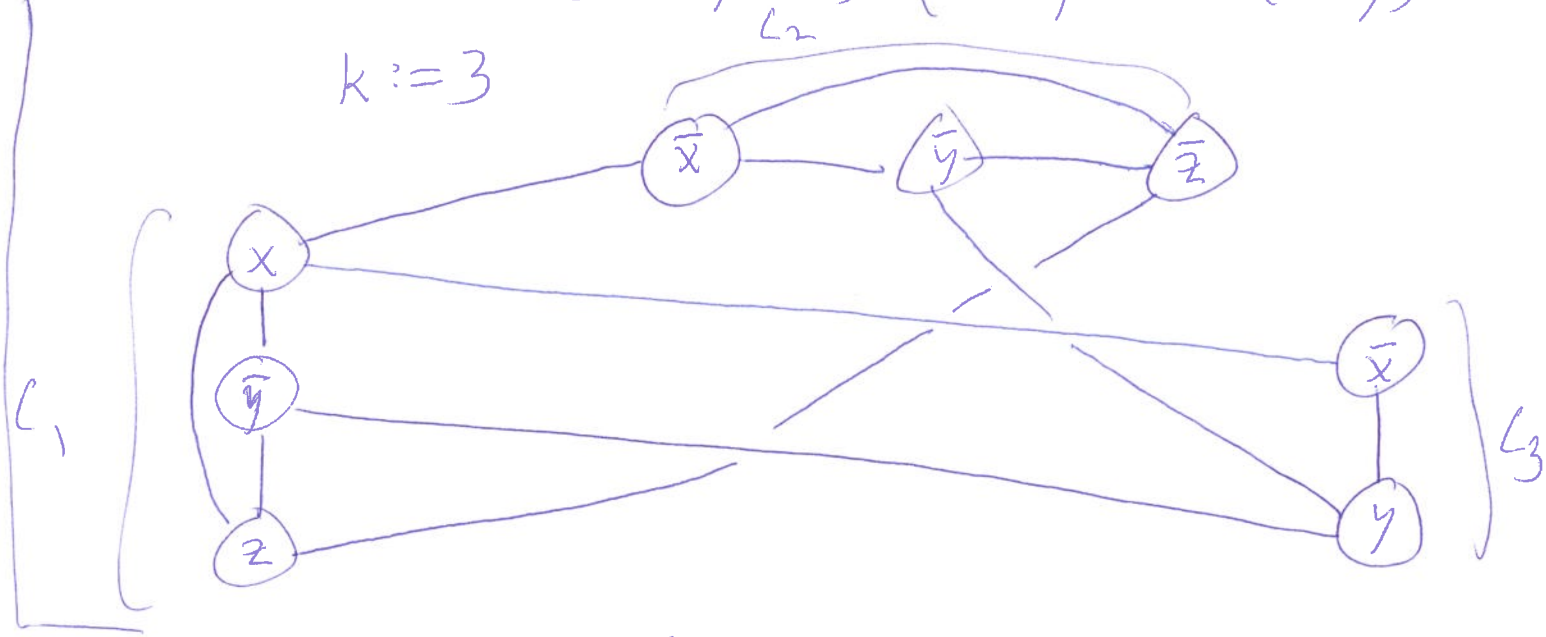
- $k' := k$ (the number of clause of φ)
- G has a vertex for each occurrence of a literal in φ
- ~~Ess~~ Edges: vertices ℓ_1 and ℓ_2 are adjacent if ℓ_1 & ℓ_2 occur in the same clause, OR $\ell_1 = x$ and $\ell_2 = \bar{x}$ for some variable x (ℓ_1 & ℓ_2 are contradictory)

End of construction: Output $\langle G, k \rangle$

Example: If

$$\varphi = \overbrace{(x \vee \bar{y} \vee z)}^{C_1} \wedge \overbrace{(\bar{x} \vee \bar{y} \vee \bar{z})}^{C_2} \wedge \overbrace{(\bar{x} \vee y)}^{C_3}$$

$k := 3$



This reduction is clearly polynomial time.

Correctness: Note first that any indep set of G has size $\leq k$, and if the set has size k, it has exactly one vertex in each clause.

no ~~two~~ vertices of the i.s. are in the same clause.

First: WTS φ satisfiable $\Longrightarrow \exists$ i.s. of size k:

Assume φ has a satisfying assignment: $\vec{a}$

Pick one ~~true~~ literal made true by $\vec{a}$ in each clause to form a set $I \subseteq G.V$ of size k.

Claim: I is an indep. set.

1) no two vertices in the same clause b/c

2) no contradictory literals in I (all are made true by $\vec{a}$.

$\therefore$ I is an indep set of size k. //

Conversely: $\exists$ i.s. of size k $\Rightarrow$ φ is satisfiable: ③

WTS

Let J be an indep. set of size k in G.

J has exactly one literal in each clause.

Choose a truth assignment that makes each literal in J true:

For every var x occurring in φ:

1) If J contains literal x, then set $x := 1$.

mutually exclusive b/c no contradictory literals in J

2) If J contains literal $\bar{x}$, then set $x := 0$.

3) If J has neither x nor $\bar{x}$ then set $x := 1$ (arbitrary)

No contradiction between (1) & (2), so this set each literal in J to 1.

$\therefore$ Every clause contains a true literal, since J has a vx in every clause

$\therefore$ φ is satisfied by this assignment. // $\blacksquare$

CLIQUE: ~~...~~

Instance: A graph G and a number k.

Question: Does G have a clique of size k?

Prop: $IS \leq_p$ CLIQUE

[Cor: CLIQUE is NP-hard, but CLIQUE is clearly in NP, so CLIQUE is NP-complete]

Proof of the prop:

Given a graph G, and number k, output $\langle G', k \rangle$ where $G'.V = G.V$ and any two ~~distinct~~ distinct vertices in G' are adjacent iff they are not adjacent in G.

[G' is often called the complement of G, written G^c]

Since a clique in G' is an indep. set in G and vice versa,

G has an indep set of size $k \iff G'$ has a clique of size k.

This transformation is clearly p-time. $\therefore IS \leq_p$ CLIQUE $\blacksquare$

Prop: $IS \leq_p VC$ [Cor: VC is NP-complete.]

Proof: Given G & k, note that $C \subseteq G.V$ is a vertex cover iff $G.V - C$ is an indep set.

Thus G has an indep set of size $\geq k$ iff G has an vertex cover of size $\leq |G.V| - k$.

Just output $\langle G, |G.V| - k \rangle$. //

Prop: HC is NP-complete.

Proof: Reduction from VC omitted.

HAMILTONIAN PATH (HP):

Instance: A graph G with ≥ 2 vertices

Question: Is there a path in G that touches each vertex exactly once? (including endpoints)

Prop: $HC \leq_p HP$ [$\therefore$ HP is NP-complete b/c $HP \in NP$]

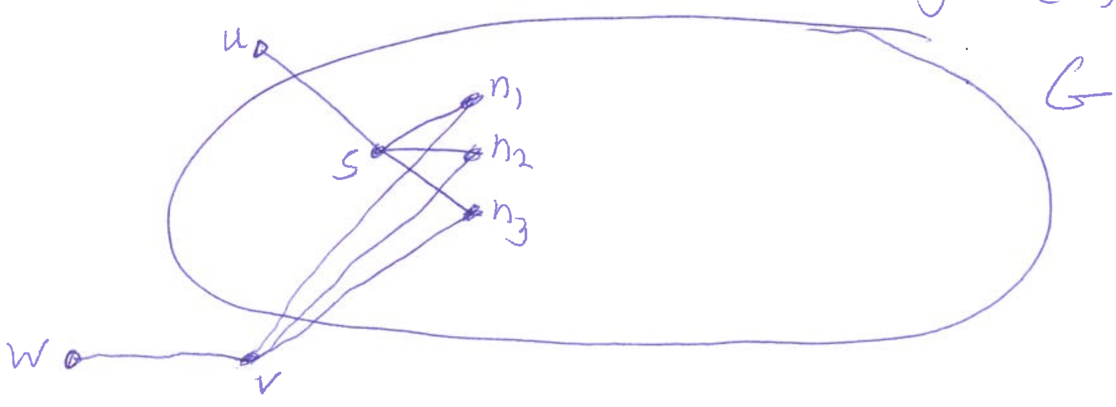
Proof: Given a graph G, choose an arbitrary vertex $s \in G.V$ not isolated. (If all vertices are isolated, then return G)

Starting with G:

1.) Add a new vertex u and new vertices $v_1, \ldots, v_k$, one for each neighbor of s (s has k neighbors) and a new vertex v.

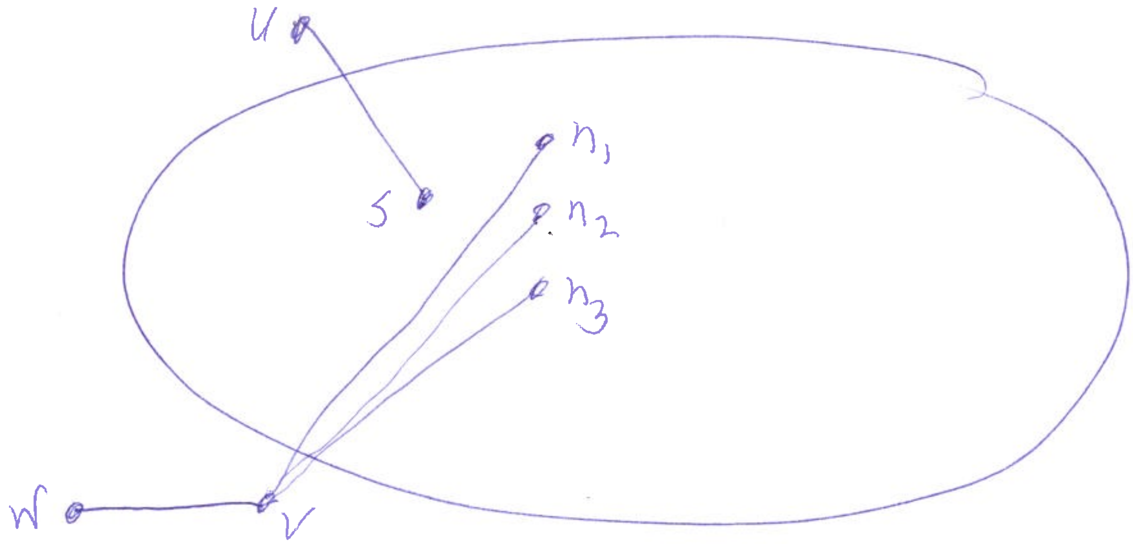
2) ~~mak~~ Add edge (s,u). For each neighbor n_j of s, add edge (n_j, v).

3) Add new vertex w and edge (v,w).



6

4) Remove the edges from s to its neighbors in G.



Call this graph G'. Output G'.

WTS: G has a H.C. iff G' has a H.P.

($\Longrightarrow$) Assume G has a H.C. c.

c visits s then one of s's neighbors through some edge (s, n_j) (some j)

then remove this edge from c,

and add ~~eges~~ edges (s,u), (n_j, v), and (v,w).

This is a H.P in G'

($\Longleftarrow$) If G' has an H.P. p, then endpoints of p must be u and w, ~~Let~~ and go through some n_j. Replace (v,w), (v,n_j), (s,u) in p with (s,n_j) to get a H.C. in G.