(114)

- Sps $\bigwedge_{i\le n} c_i$ is a 3-SAT problem (each c_i of form ~~xxxxx~~ ~~xxxx~~ $x_{i_1} \vee x_{i_2} \vee x_{i_3}$
- Define a graph G
  - Vertex set is $\{x_{i,j} : i\le n, j\le 3\}$ (i.e. each vertex represents jth var/negated var from ith clause)
  - Connect $x_{i,j}$ to $x_{i',j'}$ iff $i\neq i'$ and $x_{i,j} \neq \neg x_{i',j'}$ (i.e. connect vertices if from distinct clauses, and, as a pair, can be true simultaneously)

- Sps $\bigwedge_{i\le n} c_i$ is satisfiable by truth assn't t
- For each i, choose x_{i,j_i} s.t. $t(x_{i,j_i}) = T$
- Clearly $\{x_{i,j_i} : i\le n\}$ is a copy of K_n
- Conversely, if G contains a copy of K_n, then for each $x_{i,j}$ in this copy, define $t(x_{i,j}) = T$
- then t satisfies $\bigwedge_{i\le n} c_i$ since each $x_{i,j}$ in K_n must come from a distinct clause (no edges internal to any clause)

✓

Problem (GRAPH COLORING) (GC)
Given a graph G and $n \in \mathbb{N}$, decide whether G admits an n-coloring (i.e. if we can color vertices of G w/ n-many colors s.t. no two adjacent vertices are colored same color)

Prop'n GC is NP-complete.

Pf: - Clear again GC is NP (why)
- We reduce 3-SAT to GC
- Sps $\bigwedge_{i \le n} c_i$ is a 3-SAT prob.
- Sps the variables appearing in $\bigwedge c_i$ are $u_1, \dots, u_k$
- wlog may assume $k \geq 4$ (can add trivial new clauses if need be)
- We define a graph G that has a $(k+1)$-coloring iff $\bigwedge_{i \le n} c_i$ is satisfiable.

- Vertex set of G is
$$V = \{u_1, \dots, u_k, \neg u_1, \dots, \neg u_k, v_1, \dots, v_k, c_1, \dots, c_n\}$$

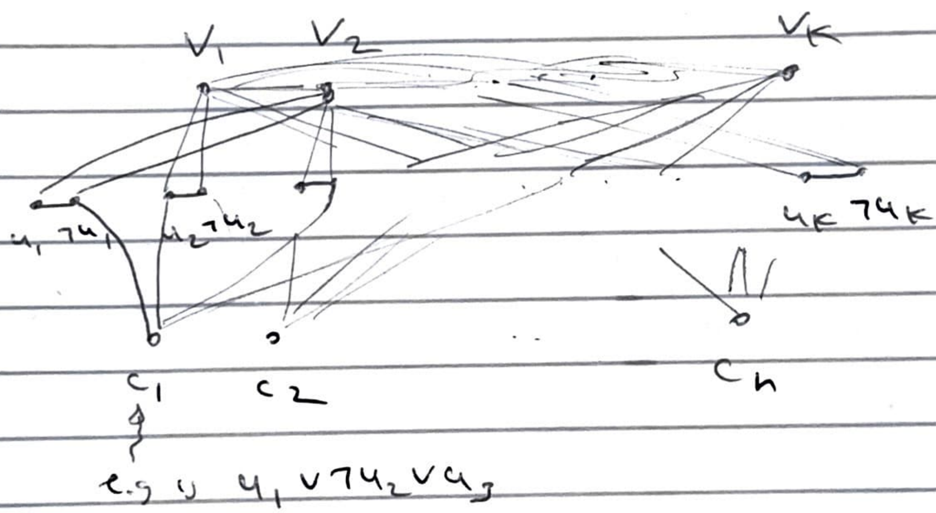
- Define edge relation E as follows:

(a.) (v_i, v_j), (v_i, u_j), (u_j, v_i), $(v_i, \neg u_j)$ $(\neg u_j, v_i)$ $\in E$ iff $i \neq j$

(b.) $(u_i, \neg u_i)$, $(\neg u_i, u_i) \in E$ $\forall i$

(c.) (u_i, c_j), $(c_j, u_i) \in E$ iff u_i does not appear as term in c_j

(d.) $(\neg u_i, c_j)$, $(c_j, \neg u_i)$ iff $\neg u_i$ not a term in c_j



- Sps $\bigwedge_{i \le n} c_i$ is satisfiable, say by a truth ass'nt t.
- We color G in $k+1$ colors
- Color each v_i w/ color i
- if $t(u_j) = T$, color u_j by j and $\neg u_j$ by $k+1$
- if $t(\neg u_j) = T$, color $\neg u_j$ by j and u_j by $k+1$
- each c_i is $x_{i1} \vee x_{i2} \vee x_{i3}$ at least one of these is assigned T

leftmost
take least such and color c_i by index of corrup. var.
(e.g. if ~~xxxxx~~ $x_{i,1}$ is T and is $\neg u_j$, color c_i by j)

- this works! Just check (a)-(d) to see no adjacent vertices get same color.

- Conversely, sps G has a $k+1$-coloring
- relabeling colors if need be, can assume v_i is colored by color i (must all be distinct colors!) (by a.)
- define a truth assignment t:
  $t(u_i) = T$ iff color of u_i is i
- notice: exactly one of $u_i, \neg u_i$ is colored i (the one assigned T), the other is $k+1$
- follows no c_j can be colored $k+1$: since $k \geq 4$, $\exists i$ s.t. neither u_i nor $\neg u_i$ in c_j and one of these colored $k+1$.
- t satisfies $\bigwedge c_i$: why: if all disjuncts in a given c_i were F then c_i adj. to a vertex of every color $\leq k$, contradiction ✓
  $\Rightarrow t$ satisfies $\bigwedge c_i$

- We state w/o proof several other NP-complete probs
- proof of NP-completeness for these is always by reduction to a known NP-complete prob.

Problem (Vertex Cover) Given a graph G and $n \in \mathbb{N}$, decide if there is a set of vertices S w/ $|S| \leq n$ s.t. every edge in G is adjacent to some $v \in S$.

(CLIQUE can be reduced to V.C.)

Problem (Hamiltonian Circuit) Given a graph $G = (V, E)$ decide whether there is a path w/o repeated vertices $v_0, v_1, \ldots, v_n = v_0$ that contains every vertex $v \in V$.

(V.C. can be reduced to H.C.)

Problem (Travelling Salesman) Given a set S of cities, an integer-valued distance function $d: S^2 \to \mathbb{N}$ and $b \in \mathbb{N}$, decide whether there is a round-trip visiting all cities in S

(H.C. can be reduced to T.S.)

The class EXP

- we define another complexity class of problems, that we can show is strictly larger than P

- Let
$$EXP = \bigcup_{c \in \mathbb{R}^+} TIME(2^{cn})$$

where $TIME(2^{cn})$ is collection of languages L for which $\exists$ a t.m. M that decides membership in L in $\leq 2^{cn}$-many steps (on an input w of length n; for all but finitely many n)

- Since for any given polynomial $p(n)$ there is $c > 0$ s.t. $p(n) \leq 2^{cn}$ $\forall n$, we have $P \subseteq EXP$
- we prove containment is strict.

- For $f: \mathbb{N} \to \mathbb{N}$ we say $g: \mathbb{N} \to \mathbb{N}$ is $O(f)$ if there is a polynomial s.t. $g(n) \leq p(f(n))$ for all (but finitely many) n.

- We say $f: \mathbb{N} \to \mathbb{N}$ is <u>proper</u> if
  - $f(n) \geq n$
  - f is non-decreasing
  - there is a t.m. M that, on input x, outputs $1^{f(|x|)}$ (i.e. M computes $f(|x|)$) and the computation takes $O(f)$-time (i.e. $\exists g \in O(f)$ s.t. M halts in $\leq g(|x|)$-many steps)

<u>Theorem</u> (Time hierarchy theorem)
Fix a proper function f. Then there is $L \in TIME(O(f)) \setminus TIME(f)$ (i.e. $\exists g \in O(f)$ s.t. $L \in TIME(g) \setminus TIME(f)$)

<u>Pf</u>: Diagonalize!
i.e.:
- Let $A = \{0, 1, (,), a, q, R, L, ;\}$ be a lang in which we can encode t.m.'s M
  - ith symbol of M encoded as $a(i)$ where (i) is i in binary
  - ith state encoded as $q(i)$

- instructions encoded as tuples
- M encoded as tuple of instructions.

Consider the following language:

$H_f = \{M; x \mid M$ (is a code for) a t.m. that halts in Q_y in $\le f(|x|)$-many steps$\}$

We claim $L = H_f$ works, i.e.

① $H_f \in TIME(O(f))$

② $H_f \notin TIME(f)$

For ① (sketch): we describe a t.m. T that checks whether $w \in H_f$ in $O(f)$ time.

- on input $M; x$, T reads instructions of M and then simulates run of M on input x up to time $f(|x|)$
- T halts in Q_y if M has halted in Q_y by this time; in Q_n if not
- "Clear" T runs in $O(f)$ time:
  - T reads $w = M; x$ (takes $O(|w|) \le O(f(|w|))$ amount of time)
  - then runs M on x for $f(|x|)$-many steps (takes $O(f(|x|)) \le O(f(|w|))$ amount of time) ✓