Reading: 8.4-8.5

Last time:

- tractability & intractability
- decision problems

Today:

- $\mathcal{NP}$-completeness
- 3-SAT $\leq_{\mathcal{P}} INDEP-SET$

A notoriously hard problem

"one problem to solve them all"

> SAT, INDEP-SET$_d$, and TSP$_d$ seem very different, what do they have in common?

Note: all example problem have short certificates that could easily verify "yes" instance.

> how would you verify??

Def: $\mathcal{NP}$ is the class of problems that have short (polynomial sized) certificates that can easily (in polynomial time) verify "yes" instances.

Historical Note: $\mathcal{NP}$ = non-deterministic polynomial time

"a nondeterministic algorithm could guess the certificate and then verify it in polynomial time"

> note: definition asymmetric wrt "yes" and "no"

> unfortunately, no non-deterministic computers exist

a quantum computer is close to being non-deterministic

Note: Not all problems are in $\mathcal{NP}$.

E.g., unsatisfiability.

Def:

- Problem X is in $\mathcal{NP}$ if exists short easily-verifiable certificate.
- Problem X is $\mathcal{NP}$-hard if $\forall Y \in \mathcal{NP}$, $Y \leq_{\mathcal{P}} X$.
- Problem X is $\mathcal{NP}$-complete if $X \in \mathcal{NP}$ and X is $\mathcal{NP}$-hard.

Lemma: INDEP-SET $\in \mathcal{NP}$.

Lemma: SAT $\in \mathcal{NP}$.

Lemma: TSP $\in \mathcal{NP}$.

Goal: show INDEP-SET, SAT, TSP are $\mathcal{NP}$-complete.

Notorious Problem: NP

input:

- decision problem verifier program VP.
- polynomial $p(\cdot)$.
- decision problem instance: x

output:

- "Yes" if exists certificate c such that $VP(x, c)$ has "verified = true" at computational step $p(|x|)$.
- "No" otherwise.

Fact: NP is $\mathcal{NP}$-complete.

is NP $\in \mathcal{P}$?

Note: Unknown whether $\mathcal{P} = \mathcal{NP}$.

Note: $\leq_{\mathcal{P}}$ is transitive: if $Y \leq_{\mathcal{P}} X$ and $X \leq_{\mathcal{P}} Z$ then $Y \leq_{\mathcal{P}} Z$.

find simpler problems to reduce from

Plan:

1. NP $\leq_{\mathcal{P}} \cdots \leq_{\mathcal{P}}$ 3-SAT [[*next time*]]
2. 3-SAT $\leq_{\mathcal{P}}$ INDEP-SET
3. 3-SAT $\leq_{\mathcal{P}}$ HC $\leq_{\mathcal{P}}$ TSP

Independent Set

Recall: INDEP-SET (decision problem)

input: $G = (V, E)$, k

output: $S \subset V$

- satisfying $\forall v \in S,\ (u, v) \notin E$
- $|S| \geq k$

Lemma: INDEP-SET is $\mathcal{NP}$-hard.

Proof: (reduction from 3-SAT)

Part I: forward instance construction

convert 3-SAT instance f into INDEP-SET instance (G, k).

literal j in clause i

- vertices v_{ij} correspond to literals l_{ij}

not variables

- edges for:
  - clause (in triangle)

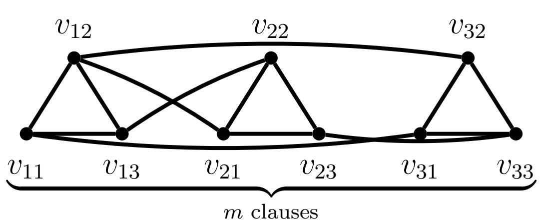
    "at most one vertex selected per clause"
  - conflicted literals.

    "vertices for conflicting literals cannot be selected"
- "vertex v_{ij} is selected" $\Rightarrow$ "literal l_{ij} is true".

converse not true!

- "indep set of size m $\Leftrightarrow$ "satisfying assignment"

Example: $f(z_1, z_2, z_3, z_4) = (z_1 \vee z_2 \vee z_3) \wedge (\bar{z}_2 \vee \bar{z}_3 \vee \bar{z}_4) \wedge (\bar{z}_1 \vee \bar{z}_2 \vee z_4)$



Runtime Analysis: linear time (one vertex per literal).

Part II:: reverse certificate construction

construct assignment $\mathbf{z}$ from S

(if G has indep. set S size $\geq m$ then f is satisfiable.)

(a) For each z_r

- if exists nodes in S are labeled by "z_r"

  $\Rightarrow$ set $z_r = 1$
- else

  $\Rightarrow$ set $z_r = 0$

Note: no two nodes $u, v \in S$ labeled by both z_r or $\bar{z}_r$, if so, there is (u, v) edge so S would not be independent.

(b) $f(\mathbf{z}) = T$:

- S has $|S| = m$

$\Rightarrow$ S has one vertex per clause.

- for caluse i:
  - if $v_{ij} \in S$ is not negated, then i is true.
  - if $v_{ij} \in S$ is negated, then i is true.

Part III:: forward certificate construction

construct independent set S from $\mathbf{z}$

(if f is satisfiable then G has indep. set size $\geq m$.)

- let S' be nodes in G corresponding to true literals.
- if more than one node in S' in same triangle drop all but one.

$\Rightarrow$ S.

- $|S| = m$.
- for all $u, v \in S$,
  - u & v not in same triangle.
  - l_u and l_v both true

    $\Rightarrow$ must not conflict

    $\Rightarrow$ no (l_u, l_v) edge in G.
  - so S is independent.