Exercise: Modify the proof to work for non-deterministic TMs.

FINE-GRAINED COMPLEXITY

Goal: Finer results than "polynomial vs. exponential"
E.g., prove that a $\Theta(n^2)$-time alg. is optimal.

Caveat: This will be model-dependent. We will assume RAM here.

Tool: Fine-grained reduction

$$\text{problem } A \xrightarrow[\text{runs in time } r(n), \text{ produces output of size } s(n)]{f} \text{problem } B$$

If B can be solved in time $T(n)$, then A can be solved in time $O(r(n) + T(s(n)))$
covers copying of input/output of f

Upper bounds for B imply upper bounds for A.
Lower bounds for A imply lower bounds for B.

Orthogonal Vectors Problem (OV):

Input: two sets of vectors $A, B \subseteq \{0,1\}^d$, $|A|, |B| \leq n$

Question: are there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$? ← i.e., bitwise AND is everywhere zero

Baseline algorithms: $O(n^2 d)$ trivial, $O(nd \cdot 2^d)$ ← for each $a \in A$, construct all orthogonal vectors and look them up in a suitable data structure for B (e.g., a trie)

Hypothesis (OVH): For no $\varepsilon > 0$, there is an algorithm solving OV in time $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$.

NFA Acceptance problem (NFAA):

Input: Non-deterministic finite-state automaton M of size $|M| = \#\text{states} + \#\text{transitions}$, string α.

Query: Does M accept α?

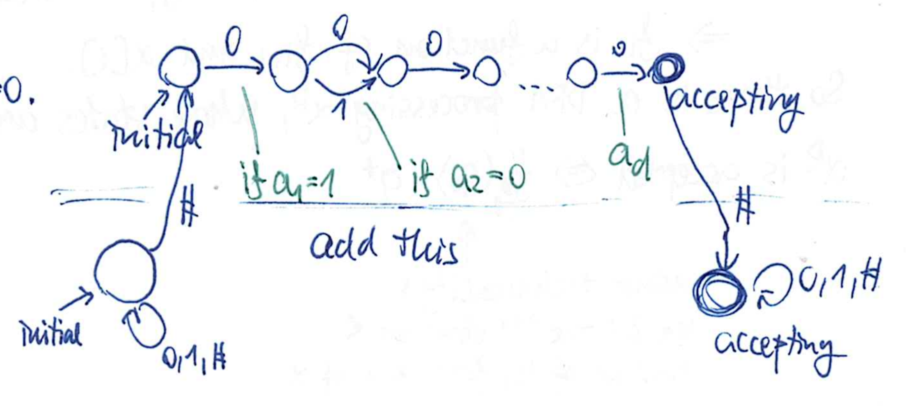
Baseline: $O(|M| \cdot |\alpha|)$ by computing δ^* (see previous lecture)
$O(2^{|M|} + |\alpha|)$ by reducing to a DFA first.

Theorem: Assuming OVH, there is no $\varepsilon > 0$ s.t. NFAA can be solved in time $O((|M| \cdot |\alpha|)^{1-\varepsilon})$.

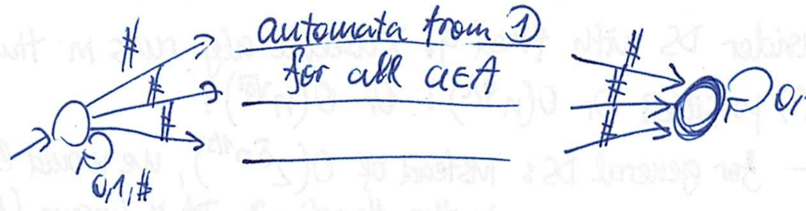
Proof: Will show reduction OV → NFAA running in time $O(nd)$, producing $|M| \in O(nd)$, $|\alpha| \in O(nd)$.
If NFAA can be solved in $O((|M| \cdot |\alpha|)^{1-\varepsilon})$ time for some $\varepsilon > 0$,
then OV can be solved in $O((n^2 d^2)^{1-\varepsilon}) = O(n^{2-2\varepsilon} \cdot d^{2-2\varepsilon})$ time, contradicting OVH.
Now the reduction:

① given $a \in A$, construct NFA which accepts $b \Leftrightarrow \langle a, b \rangle = 0$.

② given a, construct NFA accepting $\# b^1 \# b^2 \# \ldots \# b^n \#$ $\Leftrightarrow \exists i : \langle a, b^i \rangle = 0$



③ Add a choice of $a \in A$:



Surprisingly, fine-grained bounds are connected with the "big world" of P vs. NP.

<u>Exponential Time Hypothesis (ETH)</u>: ~~Formula~~ $\exists \varepsilon > 0$ s.t. 3-SAT can't be solved in $O(2^{\varepsilon N})$ time.

└ justification: baseline alg. is $O(2^N \cdot \text{poly}(M,N))$-time
state-of-the-art alg. is $O(\underbrace{1.3280^N}_{\text{improvements don't seem to converge towards 1}} \cdot \text{poly}(M,N))$-time.

for k-SAT: $N :=$ # variables, $M :=$ # clauses

Obviously, ETH implies P≠NP.

<u>Strong ETH</u> (SETH): $\forall \varepsilon > 0\ \exists k$ s.t. k-SAT cannot be solved in time $O(2^{\varepsilon N})$. ← dependence on M is not important as $M \le \binom{N}{k} \cdot 2^k$, which is polynomial for fixed k

└ justification: state-of-the-art kSAT algs are slower for higher k, converging towards 2^N.

It's known that SETH ⇒ ETH.

<u>Dominating Set problem (DS)</u>

Input: undirected graph with n vertices, $q > 0$

Question: is there a <u>dominating set</u> D of size q?

└ for $G=(V,E)$: $D \subseteq V$, $\forall u \in V\ \exists v \in D: \underbrace{(u=v \text{ or } \{u,v\} \in E)}_{u \text{ dominated by } v}$

<u>Theorem</u>: DS is NP-complete.

<u>Proof</u>: DS∈NP is trivial, will show NP-hardness by reducing 3-SAT to DS.

Will set $q=N$. This implies that a dom. set must use exactly 1 vertex from each var. gadget.

Formula satisfiable ⇒ choose dom. set according to assignment, check that all clause vertices are dominated.

Dom. set exists ⇒ choose assignment according to literals used in D ($d_i \in D$ ⇒ choose x_i arbitrarily), check that the formula is satisfied.

<u>Theorem</u>: ETH ⇒ $\exists \delta > 0$ s.t. DS cannot be solved in time $O(2^{\delta \cdot n^{1/3}})$.

<u>Proof</u>: Finer analysis of the same reduction.

Graph has n vertices, m edges for $n = 3N + M$, $m \le 3n$] ~~we~~ We have $M \le \binom{N}{3} \cdot 2^3 \le 2N^3$, so $n \le 3N^3$ for N large enough, $m \in O(n)$

Reduction runs in $O(n+m) = O(N^3)$ time.

If DS can be solved in $O(2^{\delta \cdot n^{1/3}})$ time, then 3SAT can in $O(2^{3^{1/3} \cdot \delta \cdot N})$.

For δ small enough, this contradicts the ETH.

Now consider DS with fixed q. Baseline alg. runs in time $O(\binom{n}{q}\cdot qn) \le O(n^{q+1})$.
Is $O(n^q)$ possible? Or $O(n^{q/2})$? Or $O(n^{\sqrt{q}})$?

for general DS: Instead of $O(2^{\delta n^{1/3}})$, we would like $O(2^{\delta n})$... how to get rid of N^3 in the #vertices? It is known (but we won't proved it here) that 3-SAT is hard even for sparse formulas ($M \in O(N)$).

Theorem: If ETH holds, then $\exists \delta > 0\ \forall^* q$: DS cannot be solved in $O(n^{\delta q})$ time.

Proof: We will show that a $O(n^{\delta q})$-time alg. for DS with $q \ge \frac{2}{\delta}$ (*) implies a $O(2^{\delta N})$ alg. for 3-SAT. So for δ small enough, this would contradict the ETH.
Modify the previous reduction of 3-SAT to DS:

- ~~divide~~ partition variables to q groups per N/q variables
- variable gadgets: for each group, create vertices for all partial assignments setting variables in the group $\rightarrow 2^{N/q}$ vertices
  + add an extra d_i vertex
  edges form a clique
- clause gadgets: for each clause, add vertex c_i connected to all partial assignments which satisfy this clause

Again, a DS of size q selects ~~a~~ 1 vertex from each var. gadget.
This either selects one partial assignment to vars in the group or d_i = "pick any".
Graph size: $n = q(2^{N/q}+1) + M \in O(2^{N/q})$ for fixed q
$m = (2^{N/q}+1)^2 + 3M \in O(2^{2N/q}) \overset{\text{because of (*)}}{\subseteq} O(2^{\delta N})$
Reduction takes $O(n+m) = O(2^{\delta N})$ time.
SAT is solved in $O(n^{\delta q}) \le O(2^{\frac{N}{q}\cdot \delta q}) = O(2^{\delta N})$ time.

- For hardness of OV, we will need the SETH (plain ETH is not known to suffice)

Theorem: SETH $\Rightarrow$ OVH.

Proof: Will show a reduction k-SAT $\longrightarrow$ OV
N variables, M clauses $\longrightarrow$ 2n vectors, dimension d
will produce $n = 2^{N/2}$, $d = M$ in time $O(nd)$, assuming k fixed.

So a $O(n^{2-\varepsilon} d)$-time alg. for OV implies a $O(2^{\frac{2-\varepsilon}{2}N}\cdot M)$-time alg. for k-SAT, contradicting SETH for k large enough.

Reduction: Split variables to 2 groups X, Y of size $N/2$.
Construct A using X:
- vectors correspond to partial assignments to X] $2^{N/2}$ vectors
- coordinates correspond to clauses] M coordinates
- "0" means that the clause is satisfied by the partial assgmt.

Similarly, construct B using Y.
$\langle a,b\rangle = 0 \Leftrightarrow$ all clauses are satisfied by a ~~[illegible]~~ union of the two part. assgmts.

Problem: Longest Common Subsequence (LCS) — define $L(\alpha,\beta) :=$ max. length of a common sub-sequence of α,β

Obtained by deleting elements while preserving order (i.e., not a subword)

Input: Strings $\alpha,\beta \in \Sigma^*$, $|\alpha|, |\beta| \le n$; $c > 0$

Output: Is $L(\alpha,\beta) > c$?

Theorem: LCS can be solved in time $O(n^2)$ independent of $|\Sigma|$.

Proof: Let $A[i,j] := L(\alpha[:i], \beta[:j])$.

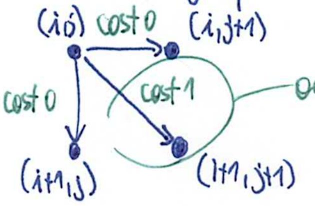
We have $A[0,j] = A[i,0] = 0$,

$$A[i,j] = \min \begin{cases} A[i-1,j] \\ A[i,j-1] \\ A[i-1,j-1]+1 \quad \text{only if } \alpha[i-1] = \beta[j-1] \end{cases} \quad \text{for } i,j>0$$

Using this, we can fill in the table of $A[i,j]$'s row by row in time $O(n^2)$.
Then $A[n,n] = L(\alpha,\beta)$.

Alternative proof: Define a directed graph with $V = \{0 - |\alpha|\} \times \{0 - |\beta|\}$, $|V| \in O(n^2)$

Edges:



$|E| \le 3|V| \in O(n^2)$

Path from $(0,0)$ to $(|\alpha|, |\beta|)$ of cost c corresponds to a common subseq. of length c.
↳ LCS = cost of the longest path ... but since the graph is acyclic, this can be computed using the same recurrence as in the previous proof. ⇒ the same $O(n^2)$-time alg.

Theorem: Assuming OVH, for no $\varepsilon > 0$ there is a $O(n^{2-\varepsilon})$-time alg. for LCS.

Proof: Omitted, see the lecture notes by Karl Bringmann.

That's all, thanks for your attention! :)