$\sim$ Assignment 5 Computational Models Spring 2022 $\sim$

Saar Barak

Problem 1

Assuming $P \neq NP$

(a) Input: Sets $A_1, ..., A_n$, and a number k.
Question: do there exist k mutually disjoint sets $A_{i1}, ..., A_{ik}$
not in $\mathcal{P}$

Proof. we can reduce IS $\leq_P$ MDS as follow. let $G(V,E)$ denote an instant of IS problem, for any $v_i \in V$ generate set A_i that contain all edges that have an end in v_i i.e $v_iv_j \in A_i \Leftrightarrow v_iv_j \in E$. Its immediate that there is IND-SET $v_{i1}, ..., v_{ik}$ size k in G iff there exist k mutually disjoint sets $A_{i1}, ..., A_{ik}$. The reduction f is poly-time computable function in $O(|V| + |E|)$, since we just need travel on G and add the elements to the sets as described above. Additionally, given a sets, we can verify if they are disjoint and there are at least k of them in poly time. Hence MDS$\notin P$ $\square$

(b) Input: Sets $A_1, ..., A_n$.
Question: do there exist 3 mutually disjoint sets A_i, A_j, A_k
in $\mathcal{P}$

claim. $3IND$[1]$\in \mathcal{P}$

Proof. First notice 3IND$\in NP$ using same verifier of IS. Given $G(V,E)$ denote $|V| = n, |E| = m$ we can label its vertex $1 \dots n$. run STCON on $\langle G, v_i, v_j \rangle$ for any i, j .now by choose any triplet from $\{1 \dots n\}^3$ and check if STCON$\langle G, v_i, v_j \rangle$ return 0 for any $i, j \in \{i, j, l\}$ in total we run $\binom{n}{2}$ times STCON save result on the tape size $\binom{n}{2}$ and check the nation for any triplet in total $\binom{n}{3}$. all the following is can be compute in poly time. hence 3-IND$\in$P

$\square$

Hence using same reduction described in (a) its holds that 3$-$IND$\leq_P$ 3$-$MDS.

(c) Input: Sets $A_1, ..., A_n$, and a number k.
Question: do there exist k sets such that $A_{i1}, ..., A_{ik}$ such that $A_i \cap A_j \neq \emptyset$
not in $\mathcal{P}$

Proof. we can reduce CLIC $\leq_P$ Q$_{(c)}$ as follow. let $G = (V,E)$ denote an instant of CLIC problem. using same proses as described at (a) we generate the sets $A_{vi}, ..., A_{vn}$. Hence if G have clic size k then exists $v_1 \dots v_k$ s.t $(v_1v_2)(v_2,v_3)\dots(v_{k-1}v_k) \in E \Leftrightarrow$ for any $e = v_iv_j$ exists $x \in A_i, A_J \Leftrightarrow A_i \cap A_j$ for any $i \neq j$. $\square$

[1]decide if given graph have independence set size 3

(d) CNF_{50v}
Input: a CNF formula ψ with at most 50 variables.
Question: does there exist an assignment that satisfies ψ
in $\mathcal{P}$

Proof. Let $\mathcal{A}$ denote algorithm that get as input CNF formula ψ. If ψ contain more then 50 variables then $\mathcal{A}$ immediately reject, otherwise its "brute force" by set any possible combination of boolean assignment for the given variables, If one of them satisfy ψ its ACCEPT, and if none of them satisfy ψ then REJECT. Since from 50 variables there are total of 2^{50} possible assignment, $\mathcal{A}$ is polynomial algorithms that decide our problem. □

(e) Input: a CNF formula ψ with at most 50 clauses.
Question: does there exist an assignment that satisfies ψ
in $\mathcal{P}$

Proof. Let ψ be a CNF formula with at most 50 clauses on n literals. Consider some algorithm $\mathcal{A}(\psi)$ that try all possible ways to choose one literal from each clause whenever $x_i, \neg x_i$ does **Not** appear together $\forall i$. If $\mathcal{A}$ find such an assignment $\Rightarrow \psi$ satisfies, otherwise it is not .Since there are at most $(2n)^{50}$ ways to choose such an assignment, $\mathcal{A}$ is polynomial algorithms that decide our problem. □

(f) Input: a 3CNF formula ψ with even number of variables.
Question: does there exist an assignment that satisfies ψ and gives $True$ for exactly one half of the variables?
not in $\mathcal{P}$

Proof. we can reduce 3SAT $\leq_P$ 3CNF_{half} as follow. Given 3SAT formula ψ that have n variable $x_1 \dots x_n$, now construct new n variable $y_1 \dots y_n$ such that each y_i correspond to the opposite value of x_i, its can done by adding to ψ 2-CNF clauses i.e

$$(x_1 \vee y_1)(\overline{x_1} \vee \overline{y_1}) \dots (x_n \vee y_n)(\overline{x_n} \vee \overline{y_n})$$

Any solutions to the formula will have half the variables with true values and half false. the following can done in poly-time with computable function just duplicate ψ variables and add the 2-CNF clauses. □

(g) Input: undirected graph G, a number k.
Question: does there exist in G a clique of size at least k or an independent set of size at least k?
not in $\mathcal{P}$

Proof. we can reduce IS $\leq_P$ CLIC$\vee$IS as follow. let $G = (V, E)$ instant of IS problem, we can construct new graph G' by adding set of additional $|V'| = |V|$ isolated vertexes. now define

$$f(\langle G = (V, E), k \rangle) = \langle G' = (V + V', E), k + |V| \rangle$$

f is polynomial. Since $k + |V| > |V|$ then G' don't have CLIC size $k + |V|$ for sure. and exist IS size k in $G \Leftrightarrow$ exist IS size $k + |V|$ in G'. Hence CLIC$\vee$IS $\notin$P □

Problem 2

(a)

For every nontrivial $L_1, L_2 \in P, L_1 \leq L_2$ **TRUE**

Let $L_1, L_2 \in P$ W.L.O.G we can generate arbitrary f that yield $L_1 \leq_P L_2$

Algorithm 1 f on input w .

Decide $w \in L_1$
If $w \in L_1$ Return any $w' \in L_2$
If $w \notin L_1$ Return any $w'' \notin L_2$

First line can done in poly-time since $L_1 \in P$, 2^{nd} holds because $L_2 \neq \emptyset$ and 3^{nd} since $L_2 \neq \Sigma^*$. All the following can done in poly-time hence f poly-time computable reduction function from $L_1 \leq_P L_2$

(b)

For every nontrivial $L_1, L_2 \in NP, L_1 \leq_P L_2$ **Equivalent to an Open Problem**

claim. *$P = NP \Leftrightarrow$ claim $2(b)$ is True*

Proof. $\Rightarrow$ if $P = NP$ any $L_1, L_2 \in NP \Rightarrow L_1, L_2 \in P \Rightarrow$ Using 2(a) any non-trivial $L_1, L_2 \leq P$ can reduce to any other $\Rightarrow L_1 \leq_p L_2$ claim 2(b) Is true.

$\Leftarrow$ Consider some non trivial $L_2 \in P$, and assume that exist non trivial $L_1 \in NP/P$. Since $P \subseteq NP$ then $L_2 \in NP \Rightarrow$ claim 2(b) yield that exists reduction s.t $L_1 \leq_P L_2$ then $L_1 \in P$ but $L_1 \in NP/P$ and we get an contradiction $\Rightarrow$ there is no exist such $L_1 \Rightarrow NP/P = \emptyset \Rightarrow NP \subseteq P \Rightarrow P = NP$ □

(c)

$L = \{0^n 1^n \mid n \in N\}$ is NPC   **FALSE**

claim. $L \in \mathcal{LOGSPACE}$

Proof. L can decided by $O(\log(n))$-space TM $\mathcal{M}$ on input w, that work as follow:

- Check that 1 is ever followed by a 0
- Count the number of 0's and 1's.
- Compare the two counters.

First step requires no working space, just moving the head.for the second we can set 2 binary counters each size $\log(n)$, and third no need extra space.
Hence $L \in \mathcal{LOGSPACE} \Rightarrow L \notin$ NPC □

(d) $\boxed{\text{There exists a language in } \mathcal{RE} \text{ that is complete w.r.t polynomial-time reductions. } \textbf{TRUE}.}$

claim. *A_{TM} is complete w.r.t polynomial-time reductions.*

Proof. consider $f(w) = \langle \mathcal{M}, w \rangle$ for any $L \in \mathcal{RE}$ and TM $\mathcal{M}$ that accept L. f computable, and able to write the encode of $\mathcal{M}, w$ in poly time. Since $w \in L \Leftrightarrow \langle \mathcal{M}, w \rangle \in A_{TM}$, f define poly-time reduction for any arbitrary $L \in \mathcal{RE}$. A_{TM} is complete w.r.t polynomial-time reductions $\square$

(e) If there exists a deterministic TM that decides SAT in time $n^{O(\log n)}$
Then every $L \in NP$ is decidable by a deterministic TM in time $n^{O(\log n)}$.
TRUE.

Proof. Let $L \in \mathcal{NP}$ and $\mathcal{M}_{SAT}$ denote D-TM that decide SAT. Since $SAT \in \mathcal{NPC}$ then exist some poly-time reduction f s.t $L \leq_P SAT$ decide SAT. Let look at d-TM $\mathcal{M}'$.

Algorithm 2 $\mathcal{M}'$ on input w .

Computes $f(w)$
Simulate $\mathcal{M}_{SAT}(f(w))$ and **Answer** like $\mathcal{M}_{SAT}$

$w \in L \Leftrightarrow \mathcal{M}_{SAT}$ accept $f(w) \Leftrightarrow f(w) \in SAT$. Since $f \in \mathcal{O}(n) \Rightarrow |f(w)| = n^k$ and $\mathcal{M}_{SAT} \in \mathcal{O}(n^{O(\log n)})$ its following that

$$\mathcal{M}'(w) = \mathcal{M}_{SAT}(f(w)) \in {}^2\mathcal{O}(\mathcal{M}_{SAT}(n^k)) = \mathcal{O}(n^k)^{O(\log n^k)} = \mathcal{O}(n^{O(\log n)})$$

$\square$

[2] Not realy sure if its the proper way to write it.

Problem 3

claim. *if $P = NP$, there exists a polynomial-time TM, that given a $3CNF$ formula ψ , outputs a satisfying assignment for ψ or indicates one does not exists.*

Proof. first notice that if $P = NP$ then $NP \subseteq P$, since $3CNF \in NP$ then $3CNF \in P$. Its following that exist polynomial-time TM $\mathcal{M}_{3CNF}$ that decide 3CNF. Now let us define TM $\mathcal{M}'$ that given a formula $\psi = \alpha(x_1, x_2 \dots x_N)$ work such that:

Algorithm 3 $\mathcal{M}'$ on input ψ .

```
check if ψ is valid 3CNF formula, otherwise Reject
Simulate M_3CNF⟨ψ⟩ and Reject if M_3CNF⟨ψ⟩ Reject
α(x_1, x_2 ... x_N) ← ψ                      ▷ α represent the logic relation w.r.t ψ
y_j ← 0     1 ≤ j ≤ N
i ← 1
while i ≤ N do
    Simulate M_3CNF⟨α(y_1, y_2 ... y_{i−1}, T, x_{i+1} ... x_N)⟩
    if M_3CNF Accept then
        y_i ← T
    else                                      ▷ M_3CNF reject
        y_i ← F
    end if
    α(x_i) ← y_i
end while
output y_i ... y_N
```

If ψ does not have an boolean satisfy assignment then $\mathcal{M}'$ indicate that at step 2. The correctness of $\mathcal{M}'$ following from the "greedy" posses, i.e before any iteration α can be satisfied. Hence by assign each time one of the veritable to T, we check if $\mathcal{M}_{3CNF}$ accept. if its accept the assignment its satisfied, and if its reject then the value F is the valid assignment. $\mathcal{M}'$ run $N + 1 \times \mathcal{M}_{3CNF}$ that is poly-time TM that output a satisfying assignment for ψ or indicates one does not exists.

□

Problem 4

We say that a polynomial reduction f is a *shrinking reduction* if there exists n_0 such that for every $x \in \Sigma^*$ such that $n_0 \leq x, |f(x)| \leq |x| - 1$. Assuming $P \neq NP$

(a) For every two nontrivial languages $A, B \in P$ there exists a *shrinking reduction* from A to B.
Prove

Proof. using Q2(a) between any non trivial $A, B \in P$ exists mapping reduction. Since both non trivial, then exists some $b \in B, \bar{b} \notin B$. Consider f s.t

$$f(w) = \begin{cases} b & \text{if } w \in A \\ \bar{b} & \text{if } w \notin A \end{cases}$$

Its immediate that $f(w) \in B \Leftrightarrow f(w) = b \Leftrightarrow w \in A$. and f define valid poly-reduction. Let us define $n_0 = \max\{|\bar{b}|, |b|\} + 1$, we can notice that f define *shrinking reduction* from A to B since

$$x \in \Sigma^* \text{ s.t } n_0 \leq |x| \qquad |f(x)| \leq \max\{|\bar{b}|, |b|\} \leq n_0 - 1 \leq |x| - 1$$

□

(b) For every two nontrivial languages $A, B \in NPC$ there exists a *shrinking reduction* from A to B.
Disprove

Proof. let A be any non trivial language s.t $A \in NPC$. By consider the following claim is true, W.L.O.G we can choose $A = B$. then exist $shrinking-reduction$ f with some n_0 such that $A \leq_P A$. First notice that there is only finite many w s.t $|w| < n_0$. We can encode them all correct answers to an algorithm $\mathcal{A}$. For given $x \in \Sigma^*$ if $|x| < n_0$ then its encoded already to $\mathcal{A}$. Since $f(x) < |x|$, when $|x| \geq n_0$ we can apply finite time of $ff\ldots(x)$ until we achieve some $|w| < n_0$. All can done in poly-time since f is poly-reduction. Hence $\mathcal{A}$ decide any non trivial $A \in NPC$ in polynomial time, its following that $P = NP$, Contradiction. □

Problem 5

The following languages are NPC:

(a)

$$EXACT3SAT = \{\varphi \in 3SAT : \text{every clause of } \varphi \text{ has exactly 3 distinct variables}\}$$

The verifier for SAT is valid poly-time verifier for EX-3SAT$\in NP$. We can reduce 3SAT$\leq_P$EX-3SAT. Given instant of SAT $\psi = C_1 \wedge C_2 \ldots C_n$ for any clause C_i, define f work as follows

- If $C = \emptyset$ i.e ψ have empty clause then it is unsatisfiable. then f return any possible combination that can generate using 3 distinct variables i.e 2^3 clauses of 3EX-SAT

$$\begin{aligned} f(\emptyset) = &(x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z) \wedge (x \vee \neg y \vee \neg z) \wedge \\ &(\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \end{aligned}$$

$\emptyset \equiv f(\emptyset)$

- $C = (x)$ when C have just one literal. Let us define f such that

$$f((x)) = (x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee \neg z)$$

  $(x) \equiv f(x)$

- $C = (x \vee y)$ when C have just 2 literal. Let us define f such that

$$f((x \vee y)) = (x \vee y \vee z) \wedge (x \vee y \vee \neg z)$$

  $(x \vee y) \equiv f((x \vee y))$

- $C = (x \vee y \vee z)$ then $f(C) = C$

Hence let $f'(\psi) = f'(C_1 \wedge C_2 \dots C_n) = f(C_1) \wedge f(C_2) \dots f(C_n)$ define poly-time reduction from 3SAT to EX-3SAT. Following that EX-3SAT$\in NPC$

(b)

$$L_2 = \{\langle M, 1^n \rangle : \text{M is a TM and there exists a string that M accepts in n steps}\}$$

Proof. First consider some $w = \langle M, 1^n \rangle$ we can verify that $w \in L_2$, by simulate M on any input size n that is[3] $n^{|\Sigma^*|} \in \mathcal{O}(n)$, and in total $\mathcal{O}(n^{|\Sigma^*|}|M|^3 n \log n) \Rightarrow L_2 \in NP$ Let L be any $L \in NP$, and V_L denote its polynomial verifier, and assume its runtime is $p(|x|)$. Any such L can reduce $L \leq_P L_2$ as follows

$$x \in L \Leftrightarrow \exists c \quad \text{s.t } (x, c) \in V_L \Leftrightarrow f(x) = \langle V_L, 1^{p(|x|)} \rangle \in L_2$$

The correctness followed by the verifier definition, and f define computable poly-reduction from any $L \in NP$, Hence $L_2 \in NPC$ □

[3] I assumed a finite alphabet