Soham Chatterjee & Rishav Gupta
Email: sohamc@cmi.ac.in & rishavg@cmi.ac.in
Course: Complexity Theory

Assignment - 2
Roll: BMC202175 & BMC202145
Date: March 25, 2023

Problem 1

Give a full proof that $AM \subseteq \Pi_2$

Solution: We will use the following steps to prove -:

- Let $L \in AM$.
- To show that $AM \subseteq \Pi_2$ we will use a similar idea as $BPP \subseteq \Pi_2 \cap \Sigma_2$.
- So if $L \in AM$ then we know that $\exists\, p(n), q(n)$ polynomials in n and a $DTIME$ machine R such that $x \in L \iff$ For a random choice of $y_1 \in \{0,1\}^{p(n)}$ $\exists\, y_2 \in \{0,1\}^{q(n)}$ such that $R(x, y_1, y_2) = 1$ with probability $\geq \frac{3}{4}$ over random y_1.
- Now with a similar approach as we used in the error reduction of BPP we will reduce the error of our AM protocol (by parallely giving multiple instances and accepting the majority) from $\frac{1}{4}$ to $\frac{1}{2^n}$.
- Let S_x be the set of all random strings $y_1 \in \{0,1\}^{p'(n)}$ such that $\forall\, y_2 \in \{0,1\}^{(q'(n))}$ for which $R(x, y_1, y_2) = 0$.(We are doing this after the error reductions.)
- Now we will shift S_x using some k vectors $u_1, u_2 \cdots u_k$ then we will show for some k, $x \in L$ then $\bigcup\limits_{i=1}^{k}(S_x + u_i)$ covers the entire $\{0,1\}^{p'(n)}$ otherwise it wont cover it .
- Let $A_x = \bigcup\limits_{i=1}^{k}(S_x + u_i)$
- We claim if $|S_x| \leq 2^{(p'(n)-n)}$ then there is no set of k vectors such that $A_x = \{0,1\}^{p'(n)}$ if $k < 2^n$
- Because $|\bigcup\limits_{i=1}^{k}(S_x + u_i)| \leq k \cdot |S| \leq 2^{(p'(n)-n)} \cdot k < 2^m$.
- $\therefore A_x \neq \{0,1\}^{p'(n)}$
- If $|S| \geq (1 - 2^{-n}) \cdot 2^{(p'(n))}$ then $\exists$ vectors $u_1, u_2 \cdots u_k$ such that $A_x = \{0,1\}^{p'(n)}$ for $k > \frac{2^{(p'(n))}}{n}$.
- For $r \in 0, 1^{p'(n)}$,we need to find the probability $r \notin A_x$.
- $r \notin S_x + u_i \iff r + u_i \notin S_x$, so let $r_i = r + u_i$, so $r \notin A_x \iff \forall i, r_i \notin S_x$.
- So now we have that $Pr(r \notin A_x) = (1 - \frac{|S_x|}{2^{(p'(n))}})^k \leq \frac{1}{2^{nk}}$.
- Now $Pr(A_x \neq \{0,1\}^{(p'(n))}) \leq \sum\limits_{j=1}^{2^{(p'(n))}} Pr(w_j \notin A_x) \leq \frac{2^{(p'(n))}}{2^{nk}}$, where the words in $\{0,1\}^{(p'(n))}$ are indexed using the $w_j's$ (follows from the union bound).
- Now if $k > \frac{2^{(p'(n))}}{n}$ then the above probability over random $\{u_1, u_2 \cdots u_k\} < 1$. So we get that for this $k, \exists$ such k vectors for which A_x cover the set .
- So choose a k such that $\frac{2^{(p'(n))}}{n} < k < 2^n$.
- Now look at the following instance of Σ_2 .$x \in L' \iff \exists\{u_1, u_2 \cdots u_k\}$ such that $\forall\, r \in \{0,1\}^{(p'(n))}$, $\forall\, y_2 \in \{0,1\}^{(q'(n))}$

$$\bigvee_{i=1}^{k} R(x, r + u_i, y_2) = 0$$

Equivalently

$$\bigvee_{i=1}^{k}((r+u_i)\in S_x)$$

- Now we claim $x \in L' \iff x \notin L$ this is because if $x \notin L$ then $|S_x| \geq (1-2^{-n})\cdot 2^{(p'(n))}$ and if $x \in L$ then $|S_x| \leq 2^{(p'(n)-n)}$, so by above argument $L' = L^c$.
- Since we found a Σ_2 formula for $L^c \implies L \in \Pi_2$.

Hence we proved $AM \subseteq \Pi_2$.

□

Problem 2

Prove that $MA \subseteq \Pi_2 \cap \Sigma_2$

Solution: We will use the following steps and motivation and results from **Problem 1** : We know that $MA \subseteq AM$ and from **Problem 1** we know $AM \subseteq \Pi_2 \implies MA \subseteq \Pi_2$.

Now we only need to show $MA \subseteq \Sigma_2$.

- For this let $L \in MA$ then we know that $\exists, p(n), q(n)$ polynomials in n and $\exists$ a $DTIME$ machine R, such that $x \in L \iff \exists\ y_1 \in \{0,1\}^{(p(n))}$ such that for a random selection of $y_2 \in \{0,1\}^{(q(n))}$ $R(x, y_1, y_2) = 1$ with probability $\geq \frac{3}{4}$.
- Now we will derive an instance of BPP from L.
- Now consider the language $\{L' :< x, y_1 >$ such that $R(x, y_1, y_2) = 1$ with probability $\geq \frac{3}{4}$ over random $y_2.\}$.
- It is easy to see that $L' \in BPP$.
- By *Sipser Gaccs Theorem* we know that $BPP \subseteq \Sigma_2$.
- Hence we know that $L' \subseteq \Sigma_2$.
- Now $x \in L$, $n = |x| \iff$, $\exists, y_1$ such that $< x, y_1 > \in L'$.
- Now let the Σ_2 form of L' be $< x, y_1 >$ such that $\exists\ u \in \{0,1\}^{(a(n))}$ such that $\forall\ v \in \{0,1\}^{(b(n))}$, $D(x, y_1, u, v) = 1$.
- Now it is easy to see that $L \in \Sigma_2$, as we will just the club the there exist quantifiers into one there exist quantifier.
- Now the Σ_2 form of L is x such that $\exists\ (y_1, u) \in \{0,1\}^{(a(n)+p(n))}$ such that $\forall\ v \in \{0,1\}^{(b(n))}$, $D(x, y_1, u, v) = 1$.

Hence we proved that $MA \subseteq \Sigma_2$. Hence we proved $MA \subseteq \Pi_2 \cap \Sigma_2$.

□

Problem 3

Let $k \leq n$. Prove that the following family $\mathcal{H}_{n,k}$ is a collection of pairwise independent function from $\{0,1\}^n \to \{0,1\}^k$: For each $k \times n$ matrix A with entries in $GF(2)$, and $b \in (GF(2))^k$, the family $\mathcal{H}_{n,k}$ contains functions $h_{A,b}(x) = Ax + b$

Solution: The proof is as follows -:

- A pair-wise disjoint hash family is a set of functions $H = \{h : P \to Q\}$ such that $\forall\ u, v \in P$ and $\forall\ x, y \in Q$, we have $Pr_h[(h(u) = x) \wedge (h(v) = y)] = \frac{1}{|Q|^2}$ where the probability is taken uniformly over $h \in H$.
- $h_{(A,b)(x)} = Ax + b$. Now fix u, v, x, y for the proof .
- We need to calculate $Pr_{(A,b)}[Au + b = x \wedge Av + b = y]$.
- Notice that for each row the calculation is independent as we take the dot product of the vector with ith row of A and then add the ith element of the vector b to get the ith row element in the vector $Ax + b$.
- So we will prove for one row the rest will follow .
- Hence we will prove that $Pr_{(a_i \in \{0,1\}^n, b_i \in \{0,1\})}[(< a_i, u > +b_i = x_i) \wedge (< a_i, v > +b_i = y_i)] = \frac{1}{4}$.
- Lets convert the probability into $Pr_{(a_i \in \{0,1\}^n, b_i \in \{0,1\})}[(< a_i, u + v >= x_i + y_i) \wedge (< a_i, v > +b_i = y_i)]$.
- For proving this we will first prove $Pr_a[< x, a >= 1] = \frac{1}{2}$ where $x \neq 0$.
- Now for a given x we have that if suppose x has k ones then number of strings for which it gives dot product as one is $\sum_{i=1}^{\lceil \frac{k}{2} \rceil} \binom{k}{2i-1} \cdot 2^{(n-k)} = 2^{(n-k)} \cdot \frac{(2^k)}{2} = 2^{(n-1)}$. (This sum arises because we can have anything in the $(n - k)$ places where the bit is 0 and $1's$ in odd number of places from the remaining k places).
- Now size of sample space will be $2^n \cdot 2 = 2^{(n+1)}$.
- Now for the event $E = [(< a_i, u+v >= x_i+y_i) \wedge (< a_i, v > +b_i = y_i)]$ where $(a_i \in \{0,1\}^n, b_i \in \{0,1\})$ to occur we will have to satisfy the first condition before $\wedge$ we get that a_i should come from a certain set of size $2^{(n-1)}$ as we proved in the last point because (u, v, x_i, y_i) are fixed ,and for the second clause to satisfy for a choice of a_i only one of b_i 0 or 1 will satisfy th second condition hence the probability of event E happening is $\frac{2^{(n-1)}}{2^{(n+1)}} = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$.
- Since each row is independent the probability $Pr_{(A,b)}[Au + b = x \wedge Av + b = y] = \frac{1}{4^k} = \frac{1}{2^{2k}} = \frac{1}{|Q|^2}$.

Hence proved.

□

Problem 4

Prove that if $P = NP$ then $EXP = NEXP$

Solution: We know $EXP \subseteq NEXP$. Let $P = NP$. Hence it is enough to show for any $L \in NEXP$, $L \in EXP$. Suppose $L \in NTIME\left(2^{n^c}\right)$ and L is accepted by a nondeterministic turing machine $\mathcal{N}$. Then the language

$$L_{pad} = \left\{\left\langle x, 1^{2^{|x|^c}}\right\rangle \mid x \in L\right\}$$

is in NP because we construct a nondeterministic turing machine M which for any input string y first checks if there is a string z such that $y = \left\langle z, 1^{2^{|z|^c}}\right\rangle$. If not then M rejects y otherwise M simulates $\mathcal{N}$ on z for $2^{|z|^c}$ steps non-deterministically and accepts if $\mathcal{N}$ accepts otherwise reject. Hence $\mathcal{L}(M) = L_{pad}$.

Now $P = NP$ and $L_{pad} \in P$. Hence there exists a deterministic turing machine $\mathcal{M}$ for which $L_{pad} \in DTIME(n^c)$. Therefore $\mathcal{M}$ take $|y|^c$ time for any input y. Since $y = \left\langle x, 1^{2^{|x|^c}} \right\rangle$ $\mathcal{M}$ takes $O\left(2^{|x|^c}\right)$ time to accepts or reject y which means $\mathcal{M}$ accepts or rejects x in $O\left(2^{|x|^c}\right)$ time. Hence L is in EXP. Therefore $NEXP = EXP$

□

Problem 5

Show that if $NP \subseteq BPP$ then $NP = RP$

Solution: Assume $NP \subseteq BPP$. Note that $RP \subseteq NP$ unconditionally, hence we just need to show that $NP \subseteq RP$. Since SAT is $NP-complete$ it is enough to show that SAT is in RP. By assumption SAT is in BPP. Let M be a probabilistic Turing machine running in polynomial time and accepting SAT with error at most $1/2^n$.

Now we will give an algorithm $\mathcal{A}$ using M for SAT

Algorithm 1 RP Algorithm for SAT

Input: φ

```
  φ' ← φ
  for k = 1, ..., n do
      x_k ← 0
      φ' ← φ'
      if M(φ') == 0 then
          x_k ← 1
      end if
  end for
  if M accepts φ with the setting of n variables then
      Accept
  else
      Reject
  end if
```

If φ is not satisfiable then no assignment of the variables satisfies φ. Hence the algorithm rejects φ with probability 1 i.e.

$$\varphi \notin SAT \implies Pr[\mathcal{A}(\varphi) = 1] = 0$$

. If φ is satisfiable. Let $\mathcal{A}$ rejects φ. Then in the n iterations M at least gave one wrong answer at some point. Now probability of M at least gives one wrong answer in n iterations is $\frac{n}{2^n} < \frac{1}{2}$. Hence

$$\varphi \in SAT \implies Pr[\mathcal{A}(\varphi) = 1] > 1 - \frac{1}{2} = \frac{1}{2}$$

Hence $SAT \in RP$. Therefore $NP = RP$

□

Problem 6

If $PSPACE$ has polynomial size circuits then, show that $PSPACE = MA$

Solution: We will follow the given steps to prove this -:

- First we will prove that $MA \subseteq PSPACE$.
- Let $L \in MA$.

- We need to describe a $PSPACE$ algorithm for L.

- For doing this we will use the fact that $L \in MA$ then $\exists, p(n), q(n)$ polynomials in n and $\exists$ a $DTIME$ machine R, such that $x \in L \iff \exists\ y_1 \in \{0,1\}^{(p(n))}$ such that for a random selection of $y_2 \in \{0,1\}^{(q(n))}$ $R(x, y_1, y_2) = 1$ with probability $\geq \frac{3}{4}$.

- Now what we will do is look over all $y_1 \in \{0,1\}^{p(n)}$ and then calculate probability of success over y_2 by keeping a counter, if it is high enough we will accept x otherwise we will keep searching for such an y_1 until we exhaust the set$\{0,1\}^{(p(n))}$ if we are not able to find such y_1 then we will reject and for each branch described by y_1 and y_2 we will reuse the space while doing so also we will clean the counter space after evaluating all possible y_2's for a given y_1.

- Now the calculation of the work space will take polynomial space and the counters will also take at most $q(n)$(since we clean the counter after every y_1's all branches computation) bits so the total space would b polynomial in n.

- Hence we proved that $MA \subseteq PSPACE$.

- Now if $MA \in P/poly$ we need to show that $PSPACE \in MA$.

- For this we will use the fact that $PSPACE = IP$, we will in fact use a more stronger result that a problem in $PSPACE$ can be simulated using an $IP - protocol$ where whatever computation the prover does is in $PSPACE$, that is the prover can be replaced by an $PSPACE$ machine.

- The stronger result follows from $IP = PSPACE$ as the prover always outputs a polynomially bounded output in every round and at every step the prover can go over all possibilities in $PSPACE$ and check if there is a string which will make the protocol accept with high probability.

- Now what we will do is in the first round $Merlin$ will send a polysized circuit family $\{C_i\}_{i=1}^{p(|x|)}$ upto some polynomially bounded length (snce in the IP-protocol the length of the messages will be bounded by some polynomial $p(|x|)$) for $Arthur$ to simulate this circuit family as the prover in the $IP - protocol$ whose prover is a $PSPACE$ machine and will accept iff at last the $IP - protocol$ simulation will accept it.

- Notice that the simulation will be polytime as the $IP - protocol$ has polynomially many rounds and we are substituting the prover by a poly sized circuit so prover's computation can be done in polytime and the verifier is running in polytime by the definition of $IP - protocol$.(The overall computation is polytime as we are doing polytime computations for polynomially many rounds .)

- Now the circuit will be polysized in length of input , now if $x \in L$ then $\exists$ a prover for which the $IP - protocol$ accepts with a high probability then $Merlin$ will send the circuit of this prover which is honest to $Arthur$, now $Arthur$ will run the $IP - protocol$ with assuming this circuit as the prover , now $Arthur$ accepts iff this $IP - protocol$ accepts. Now by the completeness of the IP protocol it will accept with a probability $\geq \frac{3}{4}$).

- If $x \notin L$ then we get that by the soundness of the $IP - protocol$ that for any prover the acceptance probability is $\leq \frac{1}{4}$, hence it will accept with a probability $\leq \frac{1}{4}$.

□