E0 224: Computational Complexity Theory **Instructor: Chandan Saha**

Lecture 20

22 October 2014 **Scribe: Abhijat Sharma**

1 Class PP(Probabilistic Poly-time)

Recall that when we define the class BPP, we have to enforce the condition that the success probability of the PTM is bounded, "strictly" away from 1/2 (in our case, we have chosen a particular value 2/3). Now, we try to explore another class of languages where the success (or failure) probability can be very close to 1/2 but still equality is not possible.

Definition 1 *A language $L \subseteq \{0,1\}^*$ is said to be in* PP *if there exists a polynomial time probabilistic turing machine M, such that*

$$Pr\{M(x) = L(x)\} > 1/2$$

Thus, it is clear from the above definition that BPP $\subseteq$ PP, however there are problems in PP that are much harder than those in BPP. Some examples of such problems are closely related to the counting versions of some problems, whose decision problems are in NP. For example, the problem of counting how many satisfying assignments exist for a given boolean formula. We will discuss this class in more details, when we discuss the complexity of counting problems.

2 Complete problems for BPP?

After having discussed many properties of the class BPP, it is natural to ask whether we have any BPP-complete problems. Unfortunately, we do not know of any complete problem for the class BPP. Now, we examine why we appears tricky to define complete problems for BPP in the same way as other complexity classes.

We know that P $\subseteq$ BPP but it is an open question whether BPP = P. Thus with respect to this open question, here is one attempt at defining BPP-completeness.

Definition 2 *(Attempt) A language $L \in$ BPP is* BPP*-complete if for every other language $L' \in$ BPP, $L' \leq_p L$ i.e L' is polynomial-time Karp-reducible to L.*

Note that, the poly-time reduction in the definition can also be replaced with implicit logspace reduction, in that case we would be considering the question whether BPP = L.

Thus, we would like to come up with a language $L \in$ BPP, such that for every other language $L' \in$ BPP, there exists a poly-time computable function f such that

$$x \in L' \Leftrightarrow f(x) \in L$$

A seemingly natural candidate for such an L is:
$L = \{ < M, x, 1^t >: M$ is a PTM that accepts x with probability at least 2/3 in t steps$\}$.
Then, if $L' \in$ BPP, there will be poly-time PTM M that decides L' with at-least 2/3 success probability, in $q(n)$ steps, where q is some polynomial and n is the input length. Suppose we define the reduction as follows:

$$\text{for every } x, f(x) = < M, x, 1^{q(|x|)} > .$$

Clearly $x \in L'$ if and only if $f(x) \in L$. Thus, the defined language L is BPP-hard.

Now, we would like to show that $L \in \mathsf{BPP}$. Think of a PTM N, that on input $< M, x, 1^t >$, simulates M on x for t steps. Now, if $< M, x, 1^t > \in L$, then clearly $Pr\{N(< M, x, 1^t >) = 1\} \geq 2/3$. But, if $< M, x, 1^t > \notin L$, then we need to have the bound $Pr\{N(< M, x, 1^t >) = 1\} \leq 1/3$, which may not always be true because $< M, x, 1^t > \notin L$ only implies that M accepts x in 1^t steps with a probability less than $2/3$, not necessarily less than $1/3$.
For example, consider M to be just a trivial poly-time PTM that makes its decisions based on a coin-toss event, which makes its success probability exactly $1/2$, and thus, for $< M, x, 1^t > \notin L$, $Pr\{N(< M, x, 1^t >) = 1\} = Pr\{M(x) = 1\} = 1/2 \not\leq 1/3$ [1].

Remarks:

1. Note that the complete above argument would be valid even in the case when we define BPP-completeness with respect to implicit logspace reduction. L would be defined in the same way as above, and it would again be proved as BPP-hard but not BPP-complete.

2. We can also proceed similarly, and define a notion of PP-completeness with respect to poly-time reduction. Note that the definition would not fail as it did for BPP, because here the error probability is allowed to be equal to $1/2$, i.e for any language $L \in \mathsf{PP}$, if a given input $x \notin L$, $Pr\{M(x) = 1\} \leq 1/2$, where M is the poly-time PTM deciding L. Thus, because of the relaxation in the probability bound, there exist well-known natural PP-complete languages[2]. For example,

$$MAJSAT = \{\phi\text{: } \phi \text{ is true for more than half of all possible assignments } x_1, x_2, ..., x_n\}$$

3 BPTIME hierarchy theorems?

Recall that a language L is said to be in $\mathsf{BPTIME}(T(n))$ iff it can be decided by a bounded-error PTM (success probability $\geq 2/3$), having a running time of $O(T(n))$. As we have previously defined hierarchy theorems for deterministic and non-deterministic space and time complexity classes, we might ask the same questions for probabilistic computation.
For example, one might wonder whether $\mathsf{BPTIME}(n)$ is a strict subset of $\mathsf{BPTIME}(n^2)$, and try to resolve this using techniques such as diagonalisation, However, this question is an open question as of now, and diagonalisation does not seem to work because of the gap between the bounds for success and error probabilities (success probability$\geq 2/3$ and error probability$\leq 1/3$), which caused issues with BPP-completeness. For example, currently, we have not been able to prove that $\mathsf{BPTIME}(n) \neq \mathsf{BPTIME}(n^{(logn)^{10}})$ [1].

Thus, it can be concluded that inspite of being defined in a very natural way, and containing many natural computational problems such as Polynomial Identity Testing, Univariate Polynomial Factoring etc, the class BPP, at times, behaves in a seemingly different manner from the other complexity classes that we have seen[1].

4 Randomized Reduction

Definition 3 *A language L_1 reduces in polynomial time to a language L_2, denoted as $L_1 \leq_r L_2$, iff there exists a poly-time PTM M, such that*

$$\textit{If } x \in L_1, \textit{ then } Pr\{M(x) \in L_2\} \geq 2/3$$
$$\textit{If } x \notin L_1, \textit{ then } Pr\{M(x) \notin L_2\} \geq 2/3$$

Note that like in earlier definitions of randomized algorithms, the constant $2/3$ can be replaced by any other value greater than $1/2$. Also, recall that we have proved earlier that the bound $Pr\{M(x) = L(x)\} \geq 1/2 + 1/n^c$ (c is a given constant), is equivalent to the much stronger error bound, $Pr\{M(x) = L(x)\} \geq 1 - 1/2^{n^d}$ for every constant $d > 0$, where n is the input length. Thus, we can make the following claim:

Claim 4 *If $L_1 \leq_r L_2$ and $L_2 \in \mathsf{BPP}$, then $L_2 \in \mathsf{BPP}$.*

Proof: Given that $L_2 \in \mathsf{BPP}$, let M_2 be the PTM that decides L_2 (with success probability$\geq 1 - 1/2^{n^d}$). And M is the PTM defining the randomized reduction from L_1 to L_2 as described in the above definition. So, we can define another PTM M_1 to decide L_1 in BPP where M_1 would just use M and M_2 as subroutines. Therefore, for a string $x \in L_1$, $Pr\{M_1(x) = 1\} \geq Pr\{M(x) \in L_2\}.Pr\{M_2(M(x)) = 1 \mid M(x) \in L_2\}$, and by the above mentioned error reduction, $Pr\{M_2(y) = L_2(y)\} \geq 1 - 1/2^{(|y|)^d}$, where d can be made a very large number, which finally gives us $Pr\{M_1(x) = 1\} \geq 2/3.(1 - 1/2^{(|x|)^d})$ when $x \in L_1$. And, by a similar argument for the case $x \notin L_1$, we can show that for all x,

$$Pr\{M_1(x) = L_1(x)\} \geq 2/3.(1 - 1/2^{(|x|)^d})$$

Thus, when d is very large, this bound satisfies the requirements for BPP and hence the claim is proved. ■

Now, we will discuss one popular example of a randomized reduction and its significant consequences.

5 Valiant Vazirani Theorem

We first define a version of the SAT problem, called Unambiguous-SAT or Unique-SAT ($USAT$).

$$USAT = \{\phi\colon \phi \in SAT \text{ and } \phi \text{ has exactly one satisfying assignment}\}$$

Now, the Valiant-Vazirani theorem defines a randomized reduction, $SAT \leq_r USAT$:

Theorem 5 *There exists a randomized poly-time algorithm M such that,*

$$\textit{If } \phi \in SAT, \textit{ then } Pr\{M(\phi) \in USAT\} \geq 1/8n$$
$$\textit{If } \phi \notin SAT, \textit{ then } Pr\{M(\phi) \notin USAT\} = 1$$

where n is the no. of variables in the boolean formula ϕ.

Observe that the above theorem implies that if there exists a poly-time algorithm to solve $USAT$, then $\mathsf{NP} = \mathsf{RP}$ [3]. Suppose that B is a deterministic algorithm that solved USAT in polynomial time. Then $B \circ M$ would be an algorithm that, if given a satisfiable formula as input, would output 1 with probability at least $1/8n$ and, if given an unsatisfiable formula as input, would output 0 with probability 1. This would be an RP algorithm for SAT. Note that the correctness probability of the first case could be made exponentially close to 1 using polynomially many independent trials of $B \circ M$ and outputting 1 if and only if at least one trial outputs 1 (one-side error reduction). The existence of such an algorithm would imply that $SAT \in \mathsf{RP}$ and thus $\mathsf{NP} = \mathsf{RP}$, as we already know that $\mathsf{RP} \subseteq \mathsf{NP}$.

This also implies that if one could develop a polynomial-time randomized algorithm for $USAT$, i.e if $USAT \in \mathsf{BPP}$, then because of the reduction, $SAT \in \mathsf{BPP}$, which implies $\mathsf{NP} \subseteq \mathsf{BPP}$, which in turn would imly that the polynomial hierarchy collapses to the second level (Karp-Lipton Theorem). Thus, it would be a good exercise to try to come up with a randomized algorithm for $USAT$ and argue why it cannot have a polynomial runtime.

Now, in order to proceed with the proof of the Valiant-Vazirani theorem, we define a special class of family of hash functions.

Definition 6 (Pairwise Independent Hash Functions) *A family of hash functions defined as, $H_{n,m} = \{h \mid h\colon \{0,1\}^n \to \{0,1\}^m\}$, is said to be pairwise independent if, $\forall x_1, x_2 \in \{0,1\}^n$ such that $x_1 \neq x_2$, and $\forall y_1, y_2 \in \{0,1\}^m$,*

$$\Pr_{h \in H_{n,m}}\{h(x_1) = y_1 \wedge h(x_2) = y_2\} = \frac{1}{2^{2m}}$$

where h is a function from H, chosen uniformly at random[4].

Applying a union bound over the above equation, fixing one of the variables to x and iterating over all possible values of the other x_2, we get

$$\Pr_{h \in H_{n,m}}\{h(x) = y\} = \sum_{y_2 \in \{0,1\}^m} \Pr_{h \in H_{n,m}}\{h(x) = y \wedge h(x_2) = y_2\} = 1/2^m$$

The above result directly implies the notion of pairwise independence, as we can think of $h(x_1) = y_1$ and $h(x_2) = y_2$ as independent events.
With this definition in mind, now let us define one such family satisfying the above properties:

$$H = \{h_{A,b} \colon h_{A,b}(x) = Ax + b\}; A \in \{0,1\}^{m \times n}, b \in \{0,1\}^m$$

where all operations are over the binary field $\mathbb{F}_2$ (addition and multiplication modulo 2). Clearly, for any given A and b, $h_{A,b}$ takes a n-dimensional column vector x, and computes a m-dimensional output vector.

Claim 7 *H is a pairwise independent family of functions*

Proof: Pick some $x_1 \neq x_2$ and some y_1, y_2, then
$Pr\{h(x_1) = y_1 \wedge h(x_2) = y_2\}$
$= Pr\{A(x_1 + x_2) = (y_1 + y_2) \wedge Ax_2 + b = y_2\}$ (Recall that all operations are modulo 2, so $\forall x \in \mathbb{F}_2, x = -x$)
$= Pr\{A(x_1 + x_2) = (y_1 + y_2)\}.Pr\{Ax_2 + y_2 = b \mid A(x_1 + x_2) = (y_1 + y_2)\}$
Note that the first term in the above product is just a system of m linear equations in boolean variables, where each equation contains the entries of a particular row of the matrix A, and thus they are independent to each other. Each equation is satisfied with probability $1/2$ and so, the first term becomes $1/2^m$. Now that the matrix A is fixed, the LHS of the second term is fixed, so only one value of b will satisfy $Ax_2 + y_2 = b$, and b has m elements of $\mathbb{F}_2$. Thus,the product evaluates to:

$$Pr\{h(x_1) = y_1 \wedge h(x_2) = y_2\} = 2^{-m}.2^{-m}.$$

■

Now, having defined the concept of pairwise independence, we now formally describe the randomized reduction of $USAT$ to SAT.

To do the reduction from SAT to USAT, choose $m \in_R \{2, .., n+1\}, A \in_R \{0,1\}^{m \times n}, b \in_R \{0,1\}^m$ (all uniformly at random), where n is the number of variables in the given SAT formula. Then the polynomial-time computable (random) reduction function will be

$$\phi(x) \underset{\text{random reduction}}{\longrightarrow} \phi'(x) = \phi(x) \wedge (h_{A,b}(x) = 0^m)$$

Note that this is a poly-time reduction and $\mid \phi' \mid$ is also polynomially bounded, as $(h_{A,b}(x) = 0^m)$ can be visualised as n homogenous linear equations in x, which can be comfortably converted to a poly-sized boolean formula in x (in $\mathbb{F}_2$, multiplication is AND, addition is XOR).
It is clear that if $\phi \notin$ SAT, then irrespective of the choice of h (meaning choice of A and b) ,$\forall x, \phi'(x) = 0$, i.e $Pr\{\phi' \notin \mathsf{USAT}\} = 1$, so one part of the theorem is true.
Now, to prove the other part, i.e if $x \in$ SAT, $Pr\{\phi' \in \mathsf{USAT}\} \geq 1/8n$, we state the following lemma:

Lemma 8 (Valiant Vazirani Lemma) *Let $H_{n,m}$ be a pair-wise independent family of hash functions and let $S \subseteq \{0,1\}^n$ be such that $2^{m-2} \leq \mid S \mid \leq 2^{m-1}$,then*

$$\Pr_{h \in_R H}\{\textit{there is a unique } x \in S \textit{ satisfying } h(x) = 0^m\} \geq 1/8.$$

for any randomly chosen function h.

Before proving the above lemma, we discuss why this lemma proves the success bound on our randomized reduction. Let the set S be the set of all satisfying assignments of ϕ, then as $m \in_R \{2, .., n+1\}$ is chosen randomly, it satisfies $2^{m-2} \leq \mid S \mid \leq 2^{m-1}$ with probabilty $1/n$. And by the lemma, if m satisfies the given inequality, probability of a unique x satisfying $h(x) = 0^m$ is at-least $1/8$, thus overall $Pr\{\phi' \in \mathsf{USAT}\} \geq 1/8n$.

Proof of Lemma 8: Let S be the set of all satisfying assignments to ϕ and suppose we have chosen a value of m, which satisfies $2^{m-2} \leq \mid S \mid \leq 2^{m-1}$, then

$$Pr\{\phi' \in \mathsf{USAT}\} = \sum_{x \in S} Pr\{x \text{ is the unique satisfying assignment for } \phi'\}$$

For a fixed $x \in S$, let E_x be the event "x is the unique satisfying assignment to ϕ'" and E_0 be the event "x satisfies ϕ'". Thus, $Pr\{E_0\} = Pr\{h(x) = 0^m\} = 1/2^m$.
Also, let for any $y \in S$ let E_y be the event "y is a satisfying assignment to ϕ'".
Then,

$$\begin{aligned} E_x &= E_0 \wedge (\bigwedge_{y \in S, y \neq x} \overline{E_y}) \\ &= E_0 \setminus \bigcup_{y \in S, y \neq x} (E_0 \cap E_y). \\ Pr\{E_x\} &\geq Pr\{E_0\} - \sum_{y \in S, y \neq x} Pr\{E_0 \wedge E_y\} \\ &\geq 2^{-m} - 2^{-2m}(\mid S \mid -1)(\text{Applying pairwise independence}). \end{aligned} \tag{1}$$

Now, continuing from earlier,

$$\begin{aligned} Pr\{\phi' \in \mathsf{USAT}\} &= \sum_{x \in S} Pr\{E_x\} \\ &= \sum_{x \in S} (2^{-m} - (\mid S \mid -1)2^{-2m}) \\ &= \mid S \mid (2^{-m} - (\mid S \mid -1)2^{-2m}) \\ &\geq 2^{m-2}(\frac{1}{2^m} - \frac{2^{m-1}-1}{2^{2m}})(\text{Substituting } 2^{m-2} \leq \mid S \mid \leq 2^{m-1}) \\ &\geq \frac{1}{4} - \frac{2^{m-1}-1}{2^{m+2}} \\ &\geq \frac{1}{4} - \frac{1}{8} + \frac{1}{2^{m+2}} \\ &\geq \frac{1}{8}. \end{aligned} \tag{2}$$

Hence, this proves the success bound of our randomized reduction, given that $\phi \in \mathsf{SAT}$, $Pr\{\phi' \in \mathsf{USAT}\} \geq 1/8n$.

References

[1] Sanjeev Arora and Boaz Barak, 2007. *Computational Complexity: A Modern Approach*, Cambridge University Press.

[2] PP(Complexity). *http://en.wikipedia.org/wiki/PP_(complexity)*, Wikipedia.

[3] L.G Valiant and V.V Vazirani, 1986. *NP is as easy as detecting unique solutions*, Theoritical Computer Science 47: 85-93.

[4] Lecture Notes: Ronitt Rubinfield, 2012. *http://people.csail.mit.edu/ronitt/COURSE/S12/handouts/lec5.pdf*, CSAIL, MIT.