CS 6810: Theory of Computing **Fall 2021**

Lecture 14: Oct 14, 2021

Lecturer: Eshan Chattopadhyay *Scribe: Oscar So*

1 BPP and the Polynomial Hierarchy

Theorem 1.1 (**Gács-Lautemann-Sipser**). *BPP* $\subseteq \Sigma_2 \cap \Pi_2 \subseteq$ *PH. In simplified words, BPP is in the second level of the polynomial hierarchy.*

Proof. Let language $L \in$ BPP. We fix some notations/variables.

- x is the input string
- $n = |x|$
- $m = \text{poly}(n)$
- r is the randomness factor $\in \{0,1\}^m$
- $k = 2m/n$

By the erorr reduction technique for BPP, we can assume the existence of a Turing Machine M satisfying the following:

$$x \in L \Rightarrow Pr_r[M(x,r) = 1] \geq 1 - 2^{-n}$$

$$x \notin L \Rightarrow Pr_r[M(x,r) = 1] < 2^{-n}$$

We define $1_x = \{r \in \{0,1\}^m : M(x,r) = 1\}$, which represents the set of random strings where machine M will accept. We want to show: $x \in L \iff \exists\ v_1, v_2, ..., v_k \in \{0,1\}^m$, such that $\forall y \in \{0,1\}^m$, $\vee_{i=1}^k [M(x, v_i \oplus y) = 1]$. This means that $\exists i$, such that $v_i \oplus y \in 1_x$. Rewriting this statement would lead to $y \in (1_x \oplus v_i) \equiv y \in \bigcup_{i=1}^{k} (1_x \oplus v_i)$.

The following claims are now direct.

Claim 1.2. *If* $x \notin L$, *for any* $v_1, ..., v_k$, $|\bigcup_{i=1}^{k} (1_x \oplus v_i)| < k * 2^{m-n} << 2^m$

Claim 1.3. *If* $x \in L$, $|1_x| \geq 2^m * (1 - 2^{-n})$

To show the existence of $v_1, \ldots, v_k$, we will be using probabilistic method. With high probability over random choices of v_i, we prove that $\bigcup_{i=1}^{k} (1_x \oplus v_i) = \{0,1\}^m$. To show this, fix some $y \in \{0,1\}^m$, and define the "bad" event $\mathcal{E}_y$: $y \notin \bigcup_{i=1}^{k} (1_x \oplus v_i) \equiv v_1, ..., v_k \notin (1_x \oplus y)$. It follows that since $v_1, \ldots, v_k$ are independently sampled, we have $\Pr[\mathcal{E}_y] \leq 2^{-nk}$, and by a union bound, $\Pr[\exists y\ \mathcal{E}_y] \leq 2^m * 2^{-nk} = 2^{-m} < 1$. This proves the existence of $v_1, \ldots, v_k$ and hence we conclude that BPP is in the second level of the polynomial hierarchy.

In fact, we believe a much stronger derandomization hold.

Conjecture 1.4 (P = BPP). *Any problem that admits an efficient randomized algorithm (BPP) can be efficiently derandomized (P).*

A strong evidence towards this conjecture comes from the line of work referred to as "Hardness vs Randomness" that shows ways of using plausible circuit lower bounds to obtain derandomization results. We note one such result.

Theorem 1.5 (Impagliazzo and Wigderson [IW97]). *If $\exists$ a language $L \in E = DTIME(2^{O(n)})$ that requires $2^{\Omega(n)}$ sized circuits to solve it, then P = BPP.*

2 Randomness in the Space-Bounded Setting

We now discuss about the use of randomness in space-bounded setting.

$$BPSPACE(S(n)) = \{L\text{: } \exists \text{ a deterministic Turing Machine } M \text{ deciding } L \text{ using } O(S(n)) \text{ space and } 2^{O(S(n))} \text{ time, such that } \forall x \in \{0,1\}^*,\ Pr_r[M(x,r) = L(x)] \geq 2/3\}$$

One can similarly define $RPSPACE(S(n))$ which as one-sided error.
One important setting is when $S(n) = O(\log(n))$. As an example, we will prove a result that places undirected connectivity in RL. Note that Reingold [Rei08] proved the much stronger result that there is a deterministic logspace algorithm for undirected connectiveity. As we will see, the randomized algorithm is much simpler compared to the more sophisitcated algorithm of Reingold.

Recall

$$UPATH = \{< G, s, t >\text{: G is an undirected graph, such that there is a path from s to t}\}$$

We prove the following.

Theorem 2.1 ([AKL$^+$79])**.** *$UPATH \in RL$.*

Proof. Given an undirected graph input G:

- s is the source node
- t is the end node

We will first transform the graph to ensure it is regular and of degree 4, and then take a random walk of $O(n^4)$ on the graph G. We accept if at some point in the walk, we reach vertex t, and reject otherwise. This algorithm can be implemented in $O(log(n))$ space because it only requires space to store the current and next vertex in the walk, along with a counter. Thus, we then want to show that if there is a path from s to t, the algorithm will accept with probability $1/\Omega(n)$.

2.1 Reducing to 4-Regular Graph

We start this proof by reducing to a 4-regular graph, maintaining connectivity, by replacing a degree d node with a cycle of length d. We also want to ensure that every vertex has a self-loop.

Claim 2.2. *There is an implicitly computable function f in log space such that it maps any input to $UPATH < G, s, t >$ to $< G', s', t' >$ where:*

- *G' is a 4-regular graph with a self-loop on each vertex while maintaining graph connectivity*

- $\exists$ *a path from* s *to* t *in* G iff $\exists$ *a path from* s' *to* t' *in* G'

For each vertex i *in* G, G' *will have* n *vertices arranged in a cycle. For each neighboring pair* i *and* j *in* G, *connect an edge at the* i^{th} *vertex from the cycle corresponding to* j *and the* j^{th} *vertex from the cycle corresponding to* i. *We then add either one or two self-loops depending on whether or not the vertex is only connected to its neighbors on the cycle or if it has a neighbor in a different cycle respectively. This would make the graph degree equal to 4.*

2.2 Explaining Iteration of $O(n^4)$ Steps

$$Hit(G) = max_{i,j} \{\text{Expected number of steps from vertex/node } i \text{ to vertex/node } j\}$$

Theorem 2.3. *For a connected d-regular, undirected graph G:*

$$Hit(G) = O(d^2 n^3 log(n))$$

Proof. To be continued next class.

References

[AKL+79] Romas Aleliunas, Richard M Karp, Richard J Lipton, László Lovász, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 218–223. IEEE Computer Society, 1979.

[IW97] Russell Impagliazzo and Avi Wigderson. P= bpp if e requires exponential circuits: Derandomizing the xor lemma. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 220–229, 1997.

[Rei08] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM (JACM)*, 55(4):1–24, 2008.