CS 6815: Lecture 17

Instructor: Eshan Chattopadhyay Scribe: Drishti Wali

25th October 2018

1 Introduction

In this lecture we begin the topic on **Derandomizing Space Bounded Computation**. We start by defining the Turing Model we would be following, and then looking at the different space and time bounded complexity classes of our interest.

2 Turing Machine Model

Definition 2.1 (**Turing Machine**). *A **Turing Machine** or TM is specified by 3 tapes - a Read only input tape, a Read-Write work tape and a Read-Write Output tape with a head on each tape indicating the cell being read/written on by the machine on that step. It is also consisting of a constant number of states which provide instructions regarding the next state to go to as well as the movement of the heads of the three tapes. There is always a start state where the machine starts from and an accept state and a reject state where the machine stops. The machine also always starts with blank Read-Write tapes and all heads at the start of the tapes*

Definition 2.2 (**Deterministic Turing Machine**). *A **DTM** is a TM where each step is deterministic, or in other words each configuration consisting of the input tape, work tape and output tapes with their heads on a specific position and the machine on a specific state has only one other configuration to move to.*

Definition 2.3 (**Non-Deterministic Turing Machine**). *An **NTM** is a TM where any step can be non-deterministic, or in other words each configuration can possibly have multiple configuration to move to.*

Definition 2.4 (**Randomized Turing Machine**). *An **RTM** is a TM where there is another tape with random bits which is a Read-only tape and the head on the tape can only move in one direction.*

Definition 2.5 (**Space Complexity**). *The space used by a TM is defined as the number of non-blank cells of it's Read-Write tapes. In other words a Turing Machine M with **space complexity** $s(n)$ is a machine where the space used by M on any input of size n bits is atmost $s(n)$.*

Definition 2.6 (**Time Complexity**). *The time used by a TM is defined as the maximum number of steps taken by the machine before it stops. A machine has **time complexity** $t(n)$ if the time taken by the machine on any input of n bits is bounded by $t(n)$*

Definition 2.7 (**Configuration Graph**). *Given a TM, M with input $x \in \{0,1\}^*$, we define the **configuration graph** $G_{M,x}$ as one with nodes as all possible configurations of the machine.*

Each node thus is a specific configuration consisting of the input tape, work tape and output tapes with their heads on a specific position and the machine on a specific state. The node v for one configuration has an edge to another w if and only if the machine would go from v to w.

Remark 2.8. *A TM M accepts the input x if and only if there exists a path in $G_{M,x}$ from the configuration with start state to the configuration nodes with accept state.*

Remark 2.9. *The configuration graph of a turning machine with space complexity $S(n)$ must have vertex set size bounded by $2^{O(S(n))}$*

3 Complexity Classes and Hierarchy

Definition 3.1 (**DSPACE**). *A language $L \subseteq \{0,1\}^*$ belongs to* **DSPACE**$(f(n))$ *if L is accepted by a DTM with space complexity $O(f(n))$*

Definition 3.2 (**NSPACE**). *A language $L \subseteq \{0,1\}^*$ belongs to* **NSPACE**$(f(n))$ *if L is accepted by an NTM with space complexity $O(f(n))$*

Definition 3.3 (**RSPACE**). *A language $L \subseteq \{0,1\}^*$ belongs to* **RSPACE**$(f(n))$ *if there exists an RTM with space complexity $O(f(n))$ such that any $x \in L$ is accepted with probability 1 and any $x \notin L$ is accepted with probability atmost $1/3$*

Definition 3.4 (**BPSPACE**). *A language $L \subseteq \{0,1\}^*$ belongs to* **BPSPACE**$(f(n))$ *if there exists an RTM with space complexity $O(f(n))$ such that any $x \in L$ is accepted with probability atleast $2/3$ and any $x \notin L$ is accepted with probability atmost $1/3$*

Definition 3.5 (**DTIME**). *A language $L \subseteq \{0,1\}^*$ belongs to* **DTIME**$(f(n))$ *if L is accepted by a DTM with time complexity $O(f(n))$*

Lemma 3.6. **DSPACE**$(s(n)) \subseteq$ **DTIME**$(2^{O(s(n))})$

Proof. This is simply because we cannot have any turing machine running for more steps than all possible configurations of the space. □

Theorem 1 (**Savitch's Theorem**).
NSPACE$(s(n)) \subseteq$ **DSPACE**$(s(n)^2)$
RSPACE$(s(n)) \subseteq$ **DSPACE**$(s(n)^2)$

Proof. The proof relies on finding a path from the start node to accept node on the configuration graph for a Turing Machine which accepts a language in **NSPACE**$(s(n))$ in space $s(n)^2$. The key idea is to use recursion to save space while iterating over all possible distances of their path length. [Kli14] □

Theorem 2 ([BCP83]). **BPSPACE**$(s(n)) \subseteq$ **DSPACE**$(s(n)^2)$

Thus the critical questions in the area of Space Bounded Computation are beliefs about the equality of the classes **BPSPACE**and **DSPACE**

Open problem 3.7.
RSPACE$(s(n)) \overset{?}{=}$ **DSPACE**$(s(n))$
BPSPACE$(s(n)) \overset{?}{=}$ **DSPACE**$(s(n))$

By a padding argument, it is enough to show the above only for $s(n) = O(\log n)$. Let **RL**:=**RSPACE**$(\log(n))$ and **L**:=**DSPACE**$(\log(n))$

Lemma 3.8. **RL**=**L** $\implies$ **RSPACE**$(s(n))$ = **DSPACE**$(s(n))$

Proof. Let $L \in$**RSPACE**$(s(n))$ and define $L' := \{x(\lambda)^{2^{c.s(|x|)-|x|}} : x \in L\}$ where λ is a new symbol not used before. Then $L' \in$ **RL** $\implies$ $L' \in$ **L** $\implies$ $L \in$ **DSPACE**$(s(n))$ ☐

Theorem 3 ([SZ99]).
BPSPACE$(s(n)) \subseteq$ **DSPACE**$(s(n)^{\frac{3}{2}})$

4 Read-Once Branching Program

A **Read Once Branching Program** or ROBP is a model to study the space bounded randomized computation. For a randomized algorithm using s amount of space and $T \leq 2^s$ amount of random bits, the width of the ROBP is $w = 2^s$ and it's length is T. It has T layers each with w nodes corresponding to the space configuration on that randomized step. Each layer has exactly two edges going to the next layer corresponding to the random bit at this layer turning 0 or 1. Every node of the first layer corresponds to a specific input node and there is assumed to be exactly one node in the last layer corresponding to the accept node. The randomized algorithm is thus modeled as a random walk from the start node on the first layer to the last year. The probability of the random walk landing on the accept node is the probability of the input string being accepted by the algorithm.

Now we can specify an ROBP using function $B_{w,T} : \{0,1\}^T \to \{0,1\}$ corresponding to the steps of the random walk on it. Now, if we have a PRG $G : \{0,1\}^r \to \{0,1\}^T$ which is computable in space s and steps T such that for any $B_{w,T}$,

$$|\mathbf{Pr}[B(U_T) = 1] - \mathbf{Pr}[B(G(U_r)) = 1]| \leq \epsilon$$

then we can accept any language which $B_{w,T}$ accepts within probability ϵ. Moreover, if $r = \log T$ then we can de-randomize the program and since every language in **RL** would have an ROBP we can de-randomize **RL**.
Several such PRGs have been given such as [NW94], [IW98] and [RRV02]. Here, we would be warming up towards the Nisan PRG of seed length $((s + \log(\frac{T}{\epsilon})).\log(\frac{T}{\epsilon}))$ by building a much weaker PRG with seed length $O(s + \log(\frac{T}{\epsilon}))$.

Theorem 4. *There exists a* $G : \{0,1\}^{\frac{T}{2}+s+O(log(\frac{T}{\epsilon}))} \to \{0,1\}^T$ *such that for any ROBP* $B_{w,T}$ *of width* $w = 2^s$ *and length* T

$$|\mathbf{Pr}[B(U_T) = 1] - \mathbf{Pr}[B(G(U_r)) = 1]| \leq \epsilon$$

Proof. We begin by using $T/2$ uniformly random bits as is to use for the first $T/2$ steps of the random walk on the ROBP. There after we use the probability distribution of landing on certain nodes along with the remaining random bits to extract the rest of $T/2$ close to uniform random bits.
Let E_v: Event that after $T/2$ uniformly random steps from the start of the ROBP we land on node v. Let X be the probability distribution of the first $T/2$ steps of the random walk. Thus,

$$H_\infty(X|E_v) = \frac{T}{2} - \log(\frac{1}{\mathbf{Pr}[E_v]})$$

Let $LOW \subseteq [w]$ such that $LOW = \{v \in [w] : \mathbf{Pr}[E_v] \leq \frac{\epsilon}{2w}\}$. Thus,

$$\mathbf{Pr}[\bigcup_{v \in LOW} E_v] \leq \frac{\epsilon}{2w}.w$$

Now we take any $(T/2 - s - 1 - \log(\frac{1}{\epsilon}), \epsilon/2)$ extractor $Ext : \{0,1\}^{T/2} \times \{0,1\}^{O(s+\log(\frac{T}{\epsilon}))} \rightarrow \{0,1\}^{T/2}$ and re-use the initial $T/2$ random bits with the remaining bits as the seed to obtain the random bits for the second half of the random walk. After the first $T/2$ uniformly random steps, we land on a node in LOW with probability $\epsilon/2$ and the probability of the output of the ROBP differing from the uniform can be bounded by 1, thus the difference in probability of the outputs of the PRG and uniform in this case is $\epsilon/2$ while in the other case the output of the extractor ensures that the probability is $\epsilon/2$ close to that of the uniform thereby bounding the total difference by ϵ $\square$

References

[BCP83] Allan Borodin, Stephen A. Cook, and Nicholas Pippenger. "Parallel Computation for Well-Endowed Rings and Space-Bounded Probabilistic Machines". In: *Information and Control* 58 (1983), pp. 113–136.

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs Randomness". In: *J. Comput. Syst. Sci.* 49.2 (Oct. 1994), pp. 149–167. ISSN: 0022-0000. DOI: 10.1016/S0022-0000(05)80043-1. URL: http://dx.doi.org/10.1016/S0022-0000(05)80043-1.

[IW98] R. Impagliazzo and A. Wigderson. "Randomness vs. Time: De-Randomization Under a Uniform Assumption". In: *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*. FOCS '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 734–. ISBN: 0-8186-9172-7. URL: http://dl.acm.org/citation.cfm?id=795664.796431.

[SZ99] Michael Saks and Shiyu Zhou. "BP$_H$SPACE $(S) \subseteq$ DSPACE $(S^{3/2})$". In: 1999.

[RRV02] Ran Raz, Omer Reingold, and Salil Vadhan. "Extracting All the Randomness and Reducing the Error in Trevisan's Extractors". In: *J. Comput. Syst. Sci.* 65.1 (Aug. 2002), pp. 97–128. ISSN: 0022-0000. DOI: 10.1006/jcss.2002.1824. URL: http://dx.doi.org/10.1006/jcss.2002.1824.

[Kli14] Riley Klinger. *Savithc's Theorem*. 2014. URL: https://courses.cs.washington.edu/courses/cse431/14sp/scribes/lec16.pdf.