

Lecture 1: Introduction

*Lecturer: Partha Mukhopadhyay**Scribe: Soham Chatterjee*

1 Reference

Books:-

- Introduction to the Theory of Computation by Michael Sipser [Sip13]
- Computational Complexity: A Modern Approach by Sanjeev Arora and Boaz Barak [AB09]
- Computational Complexity: A Conceptual Perspective by Oded Goldreich [Gol08]
- Mathematics and Computation: A Theory Revolutionizing Technology and Science by Avi Wigderson [Wig19]

Lecture Notes:-

- Madhu Sudan: 2018 and 2021
- Venkat Guruswamy: 2011 and 2009
- Luca Trevisan: 2015, 2014, 2012, 2010
- Salil Vadhan: Notes
- Prahlad Harsha: 2021, 2020, 2018, 2014, 2013, 2012, 2011
- Jay Kumar Radhakrishnan: 2004

2 Basic Classes

Note:-

All the classes in this course are subsets of decidable problems

We know for any problem P a language L_P is associated.

P := Class of problems that can be decided in deterministic polynomial time

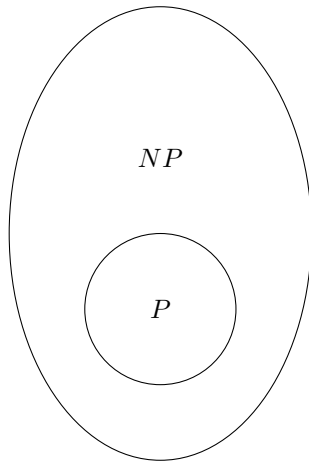
NP := Class of problems for which witness can be verified in deterministic polynomial time. Now it is obvious that P is contained in NP .

But we don't know if $P = NP$ or not.

Open Question 1

$$P \subsetneq NP$$

But it is believed that $P \neq NP$



This is called the Complexity Zoo. As the course will further progress more and more things will be added

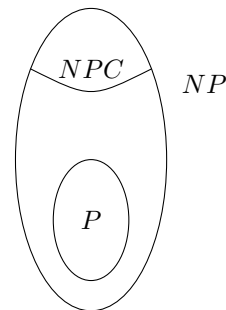
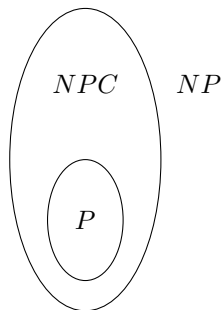
Example 1 (NP – complete (NPC))

$SAT, 3COL, VC, HAM, \dots$

$3COL :=$ Given a graph G , check whether the graph is 3-colorable

[MRG79] a has a lot of examples of NP – complete problems

Now there was question that what is the set $NP \setminus P$ if $NP \neq P$. Which one of the following true

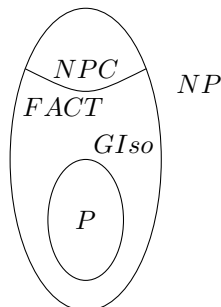


Ladner proved in [Lad75] that there exists an infinite hierarchy of problems sitting from P to NP –complete, $L \in NP \setminus (P \cup NPC)$.

But we know very less natural problems in $NP \setminus (P \cup NPC)$. Eg: Factoring (it is believed), Graph Isomorphism or $GIso$ (It was believed that $GIso$ is in $NP \setminus (P \cup NPC)$ but Laci Babai in [Bab16] brought it to very closed to P).

Primality was used to be believed in $NP \setminus (P \cup NPC)$ but later it was discovered that it is in P in [MA04]
Graph Isomorphism ($GIso$) $:= G_1 = (V_1, E_1), G_2 = (V_2, E_2)$. The question is if there exists $\phi : V_1 \rightarrow V_2$ such that $(u, v) \in E_1 \iff (\phi(u), \phi(v)) \in E_2$

Now the picture is



3 Space/Memory

Now comes another question. Are the problems in P “memory” efficient?.

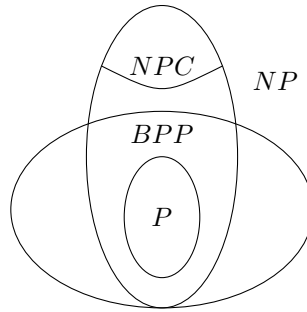
(G, E) is an undirected graph. If $u, v \in V$ then is $u \rightarrow v$ reachable? Any search algorithm like *DFS*, *BFS* works $\approx \xrightarrow{2004-Reingold[\text{Rei08}]} O(\log n)$ —memory

4 Randomness

Now we allow the Turing Machine to toss a coin to take decisions and then walk accordingly. It contains P as we can say stop tossing. Hence $P \subseteq BPP$

Here randomized algorithms are used to solve problems. Eg. [Miller Robin Primality Test](#).

Now the picture is



BPP := Class of problems which can be solved in polynomial time with the power of randomness

Open Question 2

$$P = BPP$$

But it is believed that $P = BPP$. This gives a philosophical question that can every randomized polynomial time algorithm be converted to a polynomial time deterministic algorithm with comparable time?

Note:-

BPP has no such complete problems like NP —complete problems in NP class. We have to use pseudorandom generator. So the task is to build a powerful pseudorandom generator

5 Lower Bounds

One such question on lower bound to show $P \neq NP$ is

Question 1

Let ψ be a boolean formula and $|\psi| = n$. Prove that if $\psi \in SAT$ it requires at least super linear time $(n^{1.1})$

We don't even know that

We can put some constraints or cut down some power of turing machine to show some lower bounds.

6 Diagonalization

Theorem 1

$$DTIME(n^2) \subsetneq DTIME(n^3)$$

People used to believe that these kind of ideas can be used to prove $P \neq NP$. But Natural Proofs [\[RR97\]](#) crushed this idea.

References

- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 1st edition, 2009.
- [Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.
- [Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.
- [Lad75] Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, jan 1975.
- [MA04] Nitin Saxena Manindra Agrawal, Neeraj Kayal. Primes is in p. *Annals of Mathematics*, 160 (2):781–793, 2004.
- [MRG79] David S. Johnson Michael R. Garey. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of Books in the Mathematical Sciences. W. H. Freeman, first edition edition, 1979.
- [Rei08] Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4), sep 2008.
- [RR97] Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.
- [Sip13] Michael Sipser. *Introduction to the Theory of Computation*. Cengage India Private Limited, third edition, 2013.
- [Wig19] Avi Wigderson. *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton University Press, 2019.