

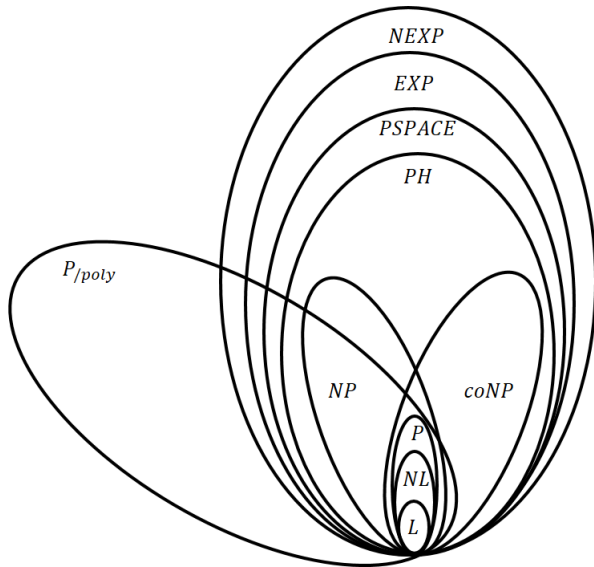
CSL853: Complexity Theory

Ragesh Jaiswal, CSE, IIT Delhi

Complexity Diagram

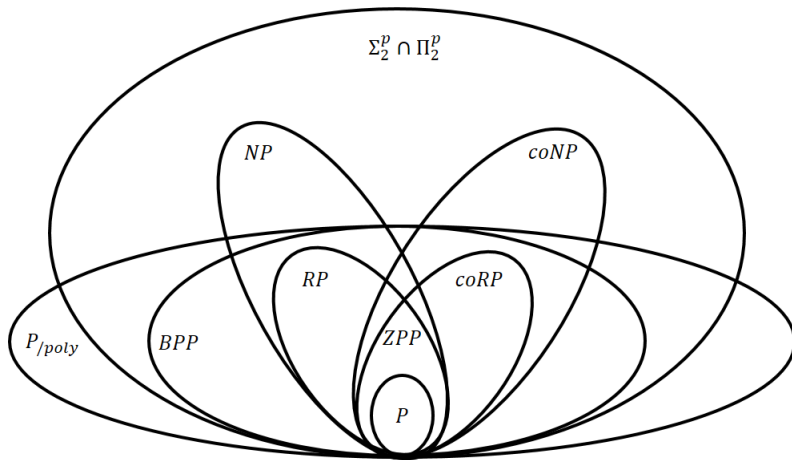
Complexity Diagram

Our current view of complexity classes



Complexity Diagram

Randomized Computation



Randomized Computation

Randomized reductions

Definition (Randomized polynomial time reduction)

Language B reduces to language C under a randomized polynomial time reduction, denoted $B \leq_r C$, if there is a PTM M such that for every $x \in \{0,1\}^*$, $\Pr[B(x) = C(M(x))] \geq 2/3$.

- Claim: If $C \in \text{BPP}$ and $B \leq_r C$, then $B \in \text{BPP}$.
- Is the relation $\leq_r$ transitive?
- We could have defined NP-completeness in terms of $\leq_r$ rather than $\leq_p$.

Complexity of Counting

Complexity of Counting

Counting problems

- The class NP captures the difficulty of finding a certificate.
- What if we are interested in *counting* the number of certificates.
- Consider the following counting problems:
 - #CYCLE: Given a directed graph, find the number of simple cycles in the graph. *The decision version of this problem is in P.*
 - #SAT: Given a Boolean formula, find the number of satisfying assignments for the formula. *The decision version of this problem is NP-complete.*
- Counting may be required for probability estimation.

Complexity of Counting

Counting problems

- Consider the following counting problems:
 - #CYCLE: Given a directed graph, find the number of simple cycles in the graph. *The decision version of this problem is in P.*
 - #SAT: Given a Boolean formula, find the number of satisfying assignments for the formula. *The decision version of this problem is NP-complete.*
- Claim: If #SAT has a polynomial time algorithm, then $P = NP$.

Theorem

If #CYCLE has a polynomial-time algorithm, then $P = NP$.

Complexity of Counting

Counting problems

Theorem

If #CYCLE has a polynomial-time algorithm, then $P = NP$.

Proof.

- Let $HAM = \{\langle G \rangle : \text{Directed graph } G \text{ has a hamiltonian cycle}\}$.
- Fact: HAM is NP-complete.
- Claim: If #CYCLE has a polynomial-time algorithm, then $HAM \in P$.
 - Given a directed graph G , construct a graph G' such that G has a hamiltonian cycle if and only if G' has at least n^{n^2} cycles.

Complexity of Counting

Counting problems

Theorem

If #CYCLE has a polynomial-time algorithm, then $P = NP$.

Proof.

- Let $HAM = \{ \langle G \rangle : \text{Directed graph } G \text{ has a hamiltonian cycle} \}$.
- Fact: HAM is NP-complete.
- Claim: If #CYCLE has a polynomial-time algorithm, then $HAM \in P$.
 - Given a directed graph G , construct a graph G' such that G has a hamiltonian cycle if and only if G' has at least n^{n^2} cycles.

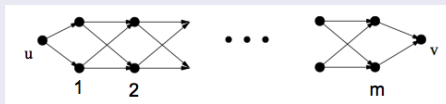


Figure : $m = n \log n$

Complexity of Counting

The class #P

Definition (#P)

A function $f : \{0, 1\}^* \rightarrow \mathbb{N}$ is in #P if there exists a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0, 1\}^*$:

$$f(x) = |\{y \in \{0, 1\}^{p(|x|)} : M(x, y) = 1\}|.$$

- #P may be alternatively defined in terms of non-deterministic machines (counting number of accepting paths).
- Let FP be the non-decisional analogue of P. That is FP contains all functions $f : \{0, 1\}^* \rightarrow \mathbb{N}$ computable by deterministic polynomial-time Turing Machines.
- Open question: Is $FP = \#P$?

Complexity of Counting

The class #P

Definition (#P)

A function $f : \{0, 1\}^* \rightarrow \mathbb{N}$ is in #P if there exists a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0, 1\}^*$:

$$f(x) = |\{y \in \{0, 1\}^{p(|x|)} : M(x, y) = 1\}|.$$

- #P may be alternatively defined in terms of non-deterministic machines (counting number of accepting paths).
- Let FP be the non-decisional analogue of P. That is FP contains all functions $f : \{0, 1\}^* \rightarrow \mathbb{N}$ computable by deterministic polynomial-time Turing Machines.
- Open question: Is $FP = \#P$?
- Claim: If $\#P = FP$, then $P = NP$.
- Open question: If $P = NP$, then does $\#P = FP$ hold?
- Claim: If $PSPACE = P$, then $\#P = P$.

Complexity of Counting

The class #P

Definition (#P)

A function $f : \{0, 1\}^* \rightarrow \mathbb{N}$ is in #P if there exists a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0, 1\}^*$:

$$f(x) = |\{y \in \{0, 1\}^{p(|x|)} : M(x, y) = 1\}|.$$

Definition (PP)

A language L is in PP if there exists a polynomial-time TM M and a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ such that for every $x \in \{0, 1\}^*$,

$$x \in L \Leftrightarrow |\{u \in \{0, 1\}^{p(|x|)} : M(x, u) = 1\}| > \frac{1}{2} \cdot 2^{p(|x|)}.$$

Theorem

If $PP = P$, then $\#P = FP$.

Complexity of Counting

#P-completeness

- Does there exist a #P-complete function?
- A machine M is said to have oracle access to a function $f; : \{0, 1\}^* \rightarrow \mathbb{N}$, if it is given access to the language $O = \{\langle x, i \rangle : f(x)_i = 1\}$.
- For a function $f : \{0, 1\}^* \rightarrow \mathbb{N}$, we define FP^f to be the set of functions that are computable by polynomial-time TM's that has oracle access to the function f .

Definition (#P-complete)

A function f is #P-complete if it is in #P and every $g \in \#P$ is in FP^f .

Theorem

#SAT is #P-complete.

Complexity of Counting

Permanent is #P-complete

- Given an $n \times n$ matrix A , the permanent of A is defined as

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^n A_{i,\sigma(i)}$$

- Note that the determinant is given by:

$$\det(A) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \prod_{i=1}^n A_{i,\sigma(i)}$$

- Claim: There is a polynomial-time algorithm for computing $\det(A)$.

Complexity of Counting

Permanent is #P-complete

- Given an $n \times n$ matrix A , the permanent of A is defined as

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^n A_{i,\sigma(i)}$$

- We will focus on 0/1 matrices.
- Given a 0/1 matrix A , how difficult is it to *determine* whether $\text{perm}(A)$ is non-zero?

Complexity of Counting

Permanent is #P-complete

- Given an $n \times n$ matrix A , the permanent of A is defined as

$$\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^n A_{i,\sigma(i)}$$

- We will focus on 0/1 matrices.
- Given a 0/1 matrix A , how difficult is it to *determine* whether $\text{perm}(A)$ is non-zero? **Can be done in polynomial-time**

Theorem (Valiant's Theorem, 1979)

perm for 0/1 matrices is #P-complete.

Complexity of Counting

Permanent is $\#P$ -complete

Theorem (Valiant's Theorem, 1979)

perm for 0/1 matrices is $\#P$ -complete.

Proof.

- We will show that $\#3SAT \in FP^{perm}$.

Complexity of Counting

Permanent is #P-complete

Theorem (Valiant's Theorem, 1979)

perm for 0/1 matrices is #P-complete.

Proof.

- We will show that $\#3\text{SAT} \in \text{FP}^{\text{perm}}$.
- Claim 1: Given any $n \times n$ matrix A , let G_A be a directed graph with adjacency matrix A . $\text{perm}(A)$ is equal to the sum of *weight of "cycle covers" of G_A* .

Complexity of Counting

Permanent is #P-complete

Theorem (Valiant's Theorem, 1979)

perm for 0/1 matrices is #P-complete.

Proof.

- We will show that $\#3\text{SAT} \in \text{FP}^{\text{perm}}$.
- Claim 1: Given any $n \times n$ matrix A , let G_A be a directed graph with adjacency matrix A . $\text{perm}(A)$ is equal to the sum of *weight of "cycle covers" of G_A* .
- Claim 2: Given a 3SAT formula ψ , there is a weighted (with weights $\{-1, 0, 1, 2, 3\}$), directed graph G_ψ such that $\text{perm}(G_\psi) = 4^{3m} \cdot \#(\psi)$.

Complexity of Counting

Permanent is #P-complete

Theorem (Valiant's Theorem, 1979)

perm for 0/1 matrices is #P-complete.

Proof.

- We will show that $\#3SAT \in FP^{perm}$.
- Claim 1: Given any $n \times n$ matrix A , let G_A be a directed graph with adjacency matrix A . $perm(A)$ is equal to the sum of *weight of "cycle covers" of G_A* .
- Claim 2: Given a 3SAT formula ψ , there is a weighted (with weights $\{-1, 0, 1, 2, 3\}$), directed graph G_ψ such that $perm(G_\psi) = 4^{3m} \cdot \#(\psi)$.
- Claim 2: Let $M = 2^{6m+n} + 1$. Given a 3SAT formula ψ , there is a weighted (with weights $\{0, 1\}$), directed graph G'_ψ such that $perm(G'_\psi) \pmod{M} = 4^{3m} \cdot \#(\psi)$.



End