

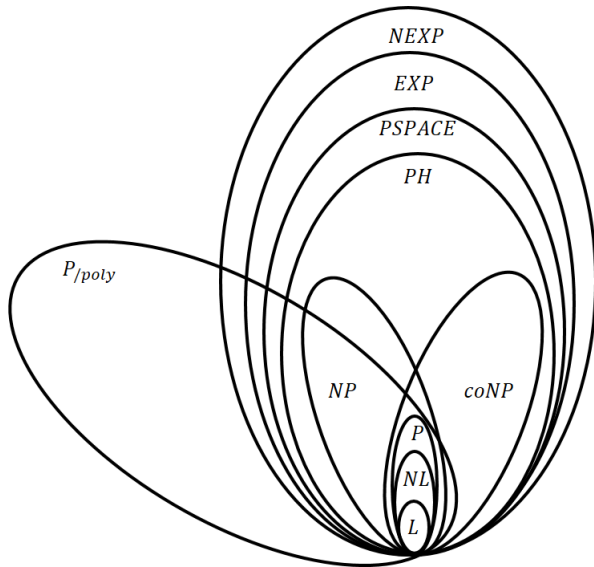
CSL853: Complexity Theory

Ragesh Jaiswal, CSE, IIT Delhi

Complexity Diagram

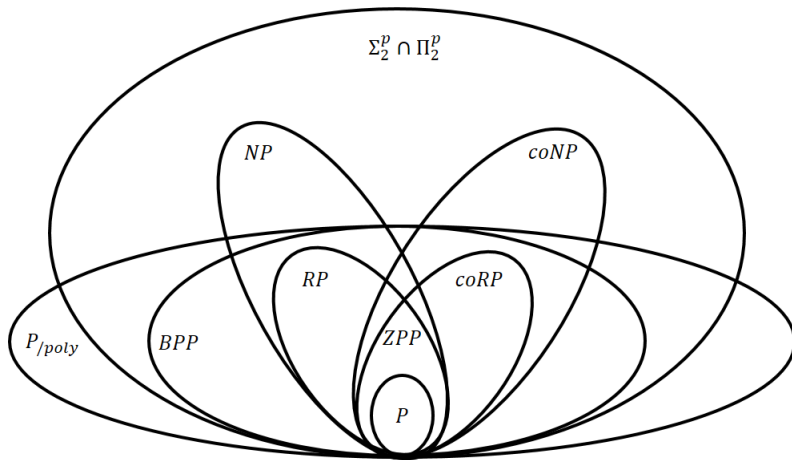
Complexity Diagram

Our current view of complexity classes



Complexity Diagram

Randomized Computation



Complexity of Counting

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 $\text{perm}(G_\psi) = 4^{3m} \cdot \#(\psi)$.
- Claim 3: 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 $\text{perm}(G'_\psi) \pmod{M} = 4^{3m} \cdot \#(\psi)$.



Complexity of Counting

Permanent is #P-complete

Claim

Given a 3SAT formula ψ , there is a weighted (with weights $\{-1, 0, 1, 2, 3\}$), directed graph G_ψ such that $\text{perm}(G_\psi) = 4^{3m} \cdot \#(\psi)$.

Proof.

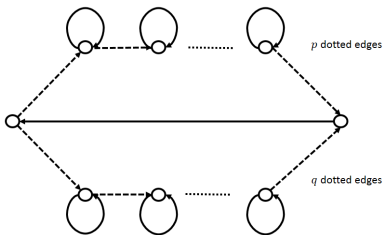
- Given ψ , we will construct G_ψ .
- We will use the following three types of “gadgets”:
 - Variable gadget
 - Clause gadget
 - “XOR” gadget

These are discussed in the next few slides.

Complexity of Counting

Permanent is #P-complete: Variable gadget

- There is one such gadget for each variable. So, n gadgets in total.
- Let variable x_1 appear in p clauses in the positive form and q clauses in the negated form.
- For every dotted edge (u, u') , it will be later removed and replaced with edges (u, v) and (v', v) for appropriately chosen vertices v, v' .

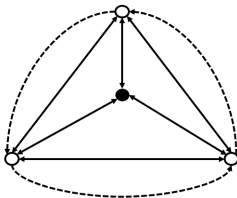


- Properties of the variable gadget:
 - Every cycle cover either takes the top loop and bottom self loops or vice versa.

Complexity of Counting

Permanent is #P-complete: Clause gadget

- There is one such gadget for each clause. So, m gadgets in total.
- For every dotted edge (u, u') , it will be later removed and replaced with edges (u, v) and (v', v) for appropriately chosen vertices v, v' .
- Each of the three dotted edges correspond to the three literals present in a clause.

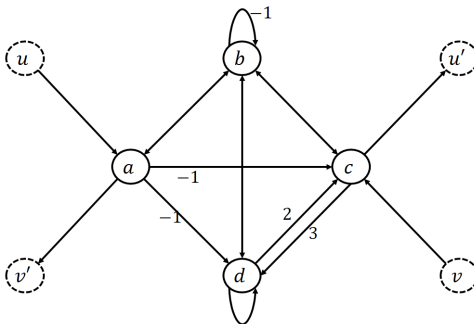


- Properties of the clause gadget:
 - 1 Any cycle cover cannot include all three dotted edges.
 - 2 For every proper subset S of dotted edges, there is a cycle cover that uses exactly edges in S .

Complexity of Counting

Permanent is #P-complete: XOR gadget

- There is one such gadget for each literal in a clause. So, $3m$ gadgets in total.



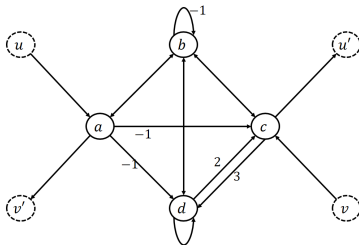
- Properties of the XOR gadget:
 - Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
 - All other cycle covers contribute 0.

Complexity of Counting

Permanent is #P-complete: XOR gadget

- Properties of the XOR gadget:

- ① Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
- ② All other cycle covers contribute 0.



- Case 1: None of $\{(u, a), (c, u'), (v, c), (a, v')\}$ is present. The cycle covers are:

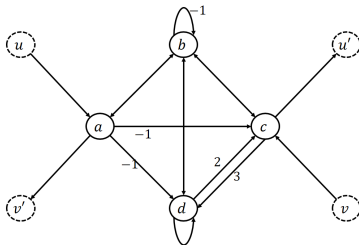
- ① $(acb)(d)$: -1
- ② $(acdb)$: -3
- ③ $(ab)(cd)$: 6
- ④ $(adcb)$: -2

Complexity of Counting

Permanent is #P-complete: XOR gadget

- Properties of the XOR gadget:

- 1 Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
- 2 All other cycle covers contribute 0.



- Case 2: $\{(u, a), (a, v')\}$ is present and $\{(c, u'), (v, c)\}$ not present.
The cycle covers are:

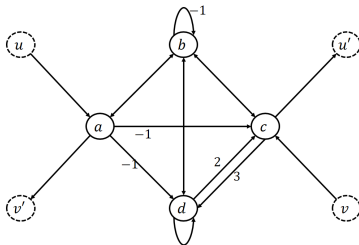
- 1 (dbc) : 3
- 2 (dcb) : 2
- 3 $(dc)(b)$: -6
- 4 $(bc)(d)$: 1

Complexity of Counting

Permanent is #P-complete: XOR gadget

- Properties of the XOR gadget:

- 1 Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
- 2 All other cycle covers contribute 0.



- Case 3: $\{(c, u'), (v, c)\}$ is present and $\{(u, a), (a, v')\}$ not present.
The cycle covers are:

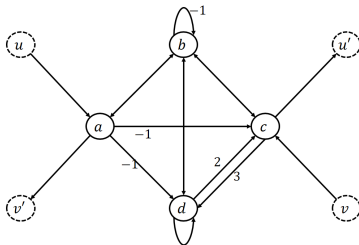
- 1 (adb) : -1
- 2 $(ab)(d)$: 1

Complexity of Counting

Permanent is #P-complete: XOR gadget

- Properties of the XOR gadget:

- 1 Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
- 2 All other cycle covers contribute 0.



- Case 4: All $\{(c, u'), (v, c), (u, a), (a, v')\}$ present. The cycle covers are:

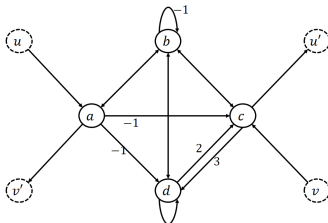
- 1 (bd) : -1
- 2 $(b)(d)$: -1

Complexity of Counting

Permanent is #P-complete: XOR gadget

- Properties of the XOR gadget:

- 1 Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
- 2 All other cycle covers contribute 0.



- Case 5: $\{(u, a), (c, u')\}$ present and $\{(v, c), (a, v')\}$ not present. The cycle covers are:

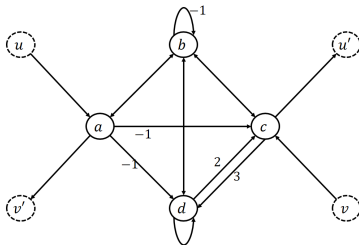
- 1 $(uacu')(bd)$: 1
- 2 $(uacu')(b)(d)$: -1
- 3 $(uabcu')(d)$: 1
- 4 $(uabdcu')$: 2
- 5 $(uadcu')(b)$: 2
- 6 $(uadbcu')$: -1

Complexity of Counting

Permanent is #P-complete: XOR gadget

- Properties of the XOR gadget:

- 1 Cycle covers that include either $\{(u, a), (c, u')\}$ or $\{(v, c), (a, v')\}$ contribute 4 to the cycle cover.
- 2 All other cycle covers contribute 0.



- Case 6: $\{(v, c), (a, v')\}$ present and $\{(u, a), (c, u')\}$ not present. The cycle covers are:

- 1 $(vcd Bav')$: 3
- 2 $(vcba)(d)$: 1

Complexity of Counting

Permanent is #P-complete

Claim

Given a 3SAT formula ψ , there is a weighted (with weights $\{-1, 0, 1, 2, 3\}$), directed graph G_ψ such that $\text{perm}(G_\psi) = 4^{3m} \cdot \#(\psi)$.

Proof.

- Given ψ , we will construct G_ψ .
- Construction: Connect the dotted edge corresponding to a literal in the clause gadget to the corresponding dotted edge in the variable gadget using an xor gadget.
- Claim: The weight of cycle covers corresponding to each satisfying assignment is equal to 4^{3m} and those corresponding to non-satisfying assignments is 0.



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 $\text{perm}(G_\psi) = 4^{3m} \cdot \#(\psi)$.
- Claim 3: 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 $\text{perm}(G'_\psi) \pmod{M} = 4^{3m} \cdot \#(\psi)$.



Complexity of Counting

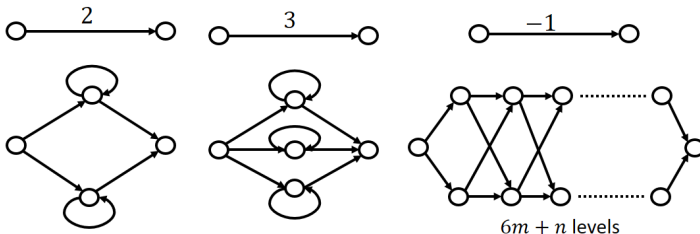
Permanent is #P-complete

Claim

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 $\text{perm}(G'_\psi) \pmod M = 4^{3m} \cdot \#(\psi)$.

Proof.

- The proof follows from the following construction.



Complexity of Counting

Approximate solutions to #P problems

Definition (α -approximation)

Let $f : \{0, 1\}^* \rightarrow \mathbb{N}$ and $\alpha < 1$. An algorithm is an α -approximation for f if for every x , $\alpha \cdot f(x) \leq A(x) \leq (1/\alpha) \cdot f(x)$.

- Approximating #CYCLE within a factor of $1/2$ is NP-hard.

Theorem (Jerrum, Sinclair, Vigoda, 2001)

*There is a **Fully Polynomial Randomized Approximation Scheme (FPRAS)** for the 0/1 permanent problem.*

Definition (FPRAS)

An (randomized) algorithm is said to be an FPRAS for some function f , if for any $1 > \delta, \epsilon > 0$, if the algorithm computes a $(1 - \epsilon)$ -approximation to the function with probability at least $(1 - \delta)$ in time $\text{poly}(n, \log 1/\delta, 1/\epsilon)$.

Complexity of Counting

$$PH \subseteq P^{\#SAT}$$

- If $\#P = FP$, then $NP = P$ and hence $PH = P$.
- If there is a $2^{\sqrt{n}}$ -time algorithm for $\#SAT$, then does it mean that any problem in PH can be solved in $2^{\sqrt{n}}$ -time?

Complexity of Counting

$$PH \subseteq P^{\#SAT}$$

- If $\#P = FP$, then $NP = P$ and hence $PH = P$.
- If there is a $2^{\sqrt{n}}$ -time algorithm for $\#SAT$, then does it mean that any problem in PH can be solved in $2^{\sqrt{n}}$ -time? **Yes.**

Theorem (Toda, 1991)

$$PH \subseteq P^{\#SAT}.$$

Complexity of Counting

$$PH \subseteq P^{\#SAT}$$

Theorem (Toda, 1991)

$$PH \subseteq P^{\#SAT}.$$

- We will need the following “parity” versions of $\#P$ and $\#SAT$ denoted by $\oplus P$ and $\oplus SAT$ respectively.

Definition ($\oplus P$)

A language L is in the class $\oplus P$ iff there is a polynomial-time non-deterministic Turing Machine M such that $x \in L$ iff the number of accepting paths of M on input x is odd.

Definition ($\oplus SAT$)

$\oplus SAT$ contain all Boolean formulas ψ such that the number of satisfying assignments of ψ is odd.

- Claim: $\oplus SAT$ is $\oplus P$ -complete.

Complexity of Counting

$$PH \subseteq P^{\#SAT}$$

Theorem (Toda, 1991)

$$PH \subseteq P^{\#SAT}.$$

Definition ($\oplus P$)

A language L is in the class $\oplus P$ iff there is a polynomial-time non-deterministic Turing Machine M such that $x \in L$ iff the number of accepting paths of M on input x is odd.

Definition ($\oplus SAT$)

$\oplus SAT$ contain all Boolean formulas ψ such that the number of satisfying assignments of ψ is odd.

- Claim 1: $\oplus SAT$ is $\oplus P$ -complete.
- Claim 2: $\oplus P \subseteq P^{\#P}$.
- $\oplus P$ may be considered the class of decision problems corresponding to the least significant bit of $\#P$ problems.
- Proof plan: Show that $PH \subseteq \oplus P$.

Complexity of Counting

$$PH \subseteq P^{\#SAT}$$

Theorem (Toda, 1991)

$$PH \subseteq P^{\#SAT}.$$

- **Proof plan:** Show that $PH \subseteq \oplus P$.
- We will start by showing that for every language $L \in NP$, there exists a language $L' \in \oplus P$ such that $L \leq_r L'$. This is known as the *Valiant-Vazirani Theorem* (1986).

Definition (Unique SAT)

$USAT = \{\psi : \psi \text{ is a Boolean formula that has a unique satisfying assignment}\}.$

Theorem (Valiant, Vazirani, 1986)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in SAT \Rightarrow \Pr[f(\psi) \in USAT] \geq \frac{1}{8n}$$

$$\psi \notin SAT \Rightarrow \Pr[f(\psi) \in SAT] = 0$$

Complexity of Counting

$$\text{PH} \subseteq \text{P}^{\#\text{SAT}}$$

Theorem (Toda, 1991)

$$\text{PH} \subseteq \text{P}^{\#\text{SAT}}.$$

- **Proof plan:** Show that $\text{PH} \subseteq \oplus\text{P}$.
- We will start by showing that for every language $L \in \text{NP}$, there exists a language $L' \in \oplus\text{P}$ such that $L \leq_r L'$. This is known as the *Valiant-Vazirani Theorem* and was proved in 1986.

Definition (Unique SAT)

$\text{USAT} = \{\psi : \psi \text{ is a Boolean formula that has a unique satisfying assignment}\}.$

Theorem (Valiant, Vazirani, 1986)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{USAT}] \geq \frac{1}{8n}$$

$$\psi \notin \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{SAT}] = 0$$

Corollary (Corollary of Valiant-Vazirani Theorem)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in \text{SAT} \Rightarrow \Pr[f(\psi) \in \oplus\text{SAT}] \geq \frac{1}{8n}$$

$$\psi \notin \text{SAT} \Rightarrow \Pr[f(\psi) \in \oplus\text{SAT}] = 0$$

Complexity of Counting

Valiant-Vazirani Theorem

Theorem (Valiant, Vazirani, 1986)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{USAT}] \geq \frac{1}{8n}$$

$$\psi \notin \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{SAT}] = 0$$

Proof.

Lemma (Valiant-Vazirani Lemma)

Let $\mathcal{H}_{n,k}$ be a family of pairwise independent hash function family from $\{0,1\}^n$ to $\{0,1\}^k$ and $S \subseteq \{0,1\}^n$ such that $2^{k-2} \leq |S| \leq 2^{k-1}$. Then

$$\Pr_{h \in \mathcal{H}_{n,k}} \left[\text{there is a unique } x \in S \text{ satisfying } h(x) = 0^k \right] \geq 1/8.$$

Complexity of Counting

Valiant-Vazirani Theorem

Theorem (Valiant, Vazirani, 1986)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{USAT}] \geq \frac{1}{8n}$$

$$\psi \notin \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{SAT}] = 0$$

Proof.

Lemma (Valiant-Vazirani Lemma)

Let $\mathcal{H}_{n,k}$ be a family of pairwise independent hash function family from $\{0,1\}^n$ to $\{0,1\}^k$ and $S \subseteq \{0,1\}^n$ such that $2^{k-2} \leq |S| \leq 2^{k-1}$. Then

$$\Pr_{h \in \mathcal{H}_{n,k}} \left[\text{there is a unique } x \in S \text{ satisfying } h(x) = 0^k \right] \geq 1/8.$$

- Given a Boolean formula ψ on n variables, pick $k \in \{2, 3, \dots, n+1\}$ and $h \in \mathcal{H}_{n,k}$ at random and consider the formula

$$\psi(x) \wedge (h(x) = 0^k).$$



Complexity of Counting

Valiant-Vazirani Theorem

Theorem (Valiant, Vazirani, 1986)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{USAT}] \geq \frac{1}{8n}$$

$$\psi \notin \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{SAT}] = 0$$

- Claim: If $\text{USAT} \in \text{P}$, then $\text{NP} = \text{RP}$.
- Open question: Is it possible to amplify the probability in the above theorem?

Complexity of Counting

Valiant-Vazirani Theorem

Theorem (Valiant, Vazirani, 1986)

There exists a polynomial-time algorithm f such that for every n -variable Boolean formula ψ

$$\psi \in \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{USAT}] \geq \frac{1}{8n}$$

$$\psi \notin \text{SAT} \Rightarrow \Pr[f(\psi) \in \text{SAT}] = 0$$

- Claim: If $\text{USAT} \in \text{P}$, then $\text{NP} = \text{RP}$.
- Open question: Is it possible to amplify the probability in the above theorem?
Not known.
- However, the following theorem can be proven.

Theorem (Randomized reduction from PH to $\oplus\text{SAT}$)

Let $c \in \mathbb{N}$ be some constant. There exists a probabilistic polynomial-time algorithm A that given a parameter m and any quantified Boolean formula ψ of size n and c levels of alternations, runs in $\text{poly}(n, m)$ time and satisfies

$$\psi \text{ is true} \Rightarrow \Pr[A(\psi) \in \oplus\text{SAT}] \geq 1 - 2^{-m}$$

$$\psi \text{ is false} \Rightarrow \Pr[A(\psi) \in \oplus\text{SAT}] \leq 2^{-m}$$

End