

Computational Complexity in Algebraic Combinatorics: recent developments

Greta Panova

University of Southern California

ICERM, September 2025

Partitions, SYTs and S_n representations

Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$$\lambda = (2, 2, 1)$$



Partitions, SYTs and S_n representations

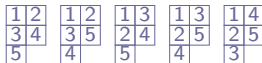
Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$$\lambda = (2, 2, 1)$$




Standard Young Tableaux of shape λ :



Partitions, SYTs and S_n representations

Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$$\lambda = (2, 2, 1)$$


Standard Young Tableaux of shape λ :

1	2	1	2	1	3	1	3	1	4
3	4	3	5	2	4	2	5	2	5
5		4		5		4		3	


Hook-length formula [Frame-Robinson-Thrall]:

$$\#\{\text{SYTs of shape } \lambda\} = f^\lambda = \frac{|\lambda|!}{\prod_{u \in \lambda} \lambda_i - i + \lambda'_j - j + 1} = \frac{5!}{4 * 3 * 2 * 1 * 1}$$

Partitions, SYTs and S_n representations

Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$$\lambda = (2, 2, 1)$$


Standard Young Tableaux of shape λ :

1	2
3	4
5	

1	2
3	5
4	

1	3
2	4
5	

1	3
2	5
4	


1	4
2	5
3	

Symmetric group S_n , its irreducible representations of S_n : the *Specht modules* \mathbb{S}_λ

Partitions, SYTs and S_n representations

Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$$\lambda = (2, 2, 1)$$


Standard Young Tableaux of shape λ :

1	2
3	4
5	

1	2
3	5
4	

1	3
2	4
5	

1	3
2	5
4	

1	4
2	5
3	

Symmetric group S_n , its irreducible representations of S_n : the *Specht modules* \mathbb{S}_λ

Example: if $V = \mathbb{C}^3$, $\pi \in S_3$, set $\pi(e_i) := e_{\pi_i}$ for $i = 1..3$, so e.g. $231 \rightarrow$

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$


$$V = \underbrace{\mathbb{C}\langle e_1 + e_2 + e_3 \rangle}_{\mathbb{S}_{(3)}} \oplus \underbrace{\mathbb{C}\langle e_1 - e_2, e_2 - e_3 \rangle}_{\mathbb{S}_{(2,1)}}$$

Partitions, SYTs and S_n representations

Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$\lambda = (2, 2, 1)$



Standard Young Tableaux of shape λ :

1	2
3	4
5	

1	2
3	5
4	

1	3
2	4
5	

1	3
2	5
4	

1	4
2	5
3	


Symmetric group S_n , its **irreducible representations** of S_n : the *Specht modules* \mathbb{S}_λ

Basis indexed by SYTs of shape λ , so $\dim \mathbb{S}_\lambda = f^\lambda := \#\{T : \text{SYT, shape } \lambda\}$.

Partitions, SYTs and S_n representations

Integer partitions and Young diagrams:

$$\lambda = (\lambda_1, \lambda_2, \dots), \lambda_1 \geq \lambda_2 \geq \dots \geq 0, \lambda_1 + \lambda_2 + \dots = n.$$

$$\lambda = (2, 2, 1)$$


Standard Young Tableaux of shape λ :

1	2
3	4
5	

1	2
3	5
4	

1	3
2	4
5	

1	3
2	5
4	

1	4
2	5
3	

Symmetric group S_n , its irreducible representations of S_n : the *Specht modules* \mathbb{S}_λ

Basis indexed by SYTs of shape λ , so $\dim \mathbb{S}_\lambda = f^\lambda := \#\{T : \text{SYT, shape } \lambda\}$.

Characters: $\text{char } \mathbb{S}_\lambda = \chi^\lambda : S_n \rightarrow \mathbb{C}$

$$\chi^{(2,1)}(231) = \text{Trace} \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} = -1$$

SSYTs, Schur functions and GL_N representations

Irreducible (polynomial) representations of the **General Linear group** $GL_N(\mathbb{C})$:

Weyl modules V_λ , indexed by highest weights λ , $\ell(\lambda) \leq N$.

Basis indexed by **Semi-Standard Young tableaux** of shape λ :

1	1	1	1	2	2	1	1	1	2	1	2
2	2	3	3	3	3	2	3	2	3	3	3

SSYTs, Schur functions and GL_N representations

Irreducible (polynomial) representations of the **General Linear group** $GL_N(\mathbb{C})$:

Weyl modules V_λ , indexed by highest weights λ , $\ell(\lambda) \leq N$.

Basis indexed by **Semi-Standard Young tableaux** of shape λ :

1	1	1	1	2	2	1	1	1	2	1	2
2	2	3	3	3	3	2	3	2	3	3	3

Schur functions: characters of V_λ : $\rho_\lambda : GL_N \rightarrow GL(V_\lambda)$

$$\mathrm{Tr}(\rho_\lambda(\mathrm{diag}(x_1, \dots, x_N))) =: s_\lambda(x_1, \dots, x_N) = \frac{\det \left[x_i^{\lambda_j + n - j} \right]_{i,j=1}^n}{\det \left[x_i^{n-j} \right]_{i,j=1}^n}$$

SSYTs, Schur functions and GL_N representations

Irreducible (polynomial) representations of the **General Linear group** $GL_N(\mathbb{C})$:

Weyl modules V_λ , indexed by highest weights λ , $\ell(\lambda) \leq N$.

Basis indexed by **Semi-Standard Young tableaux** of shape λ :

1	1	1	1	2	2	1	1	1	2	1	2
2	2	3	3	3	3	2	3	2	3	3	3

Schur functions: characters of V_λ : $\rho_\lambda : GL_N \rightarrow GL(V_\lambda)$

$$\text{Tr}(\rho_\lambda(\text{diag}(x_1, \dots, x_N))) =: s_\lambda(x_1, \dots, x_N) = \frac{\det [x_i^{\lambda_j + n - j}]_{i,j=1}^n}{\det [x_i^{n-j}]_{i,j=1}^n}$$

Generating function for **Semi-Standard Young tableaux** of shape λ :

$$s_{(2,2)}(x_1, x_2, x_3) = x_1^2 x_2^2 + x_1^2 x_3^2 + x_2^2 x_3^2 + x_1^2 x_2 x_3 + x_1 x_2^2 x_3 + x_1 x_2 x_3^2.$$

1	1	1	1	2	2	1	1	1	2	1	2
2	2	3	3	3	3	2	3	2	3	3	3

Representation theoretic multiplicities I

Tensor product of irreducible GL representations:

$$V_\mu \otimes V_\nu = \bigoplus_\lambda V_\lambda^{\oplus c_{\lambda\mu\nu}^\nu}$$

Littlewood-Richardson coefficients: $c_{\lambda\mu}^\nu$

Representation theoretic multiplicities I

Tensor product of irreducible GL representations:

$$V_\mu \otimes V_\nu = \bigoplus_\lambda V_\lambda^{\oplus c_{\lambda\mu\nu}^\nu}$$

Littlewood-Richardson coefficients: $c_{\lambda\mu}^\nu$

$$s_\mu(x)s_\nu(x) = \sum_\lambda c_{\mu\nu}^\lambda s_\lambda(x)$$

$$s_{2,1}s_{2,1} = s_{(4,2)} + s_{(4,1,1)} + s_{(3,3)} + 2s_{(3,2,1)} + s_{(3,1,1,1)} + s_{(2,2,2)} + s_{(2,2,1,1)}$$

Representation theoretic multiplicities I

Tensor product of irreducible GL representations:

$$V_\mu \otimes V_\nu = \bigoplus_\lambda V_\lambda^{\oplus c_{\lambda\mu\nu}^\nu}$$

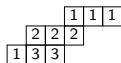
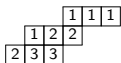
Littlewood-Richardson coefficients: $c_{\lambda\mu}^\nu$

$$s_\mu(x)s_\nu(x) = \sum_\lambda c_{\lambda\mu\nu}^\lambda s_\lambda(x)$$

$$s_{2,1}s_{2,1} = s_{(4,2)} + s_{(4,1,1)} + s_{(3,3)} + 2s_{(3,2,1)} + s_{(3,1,1,1)} + s_{(2,2,2)} + s_{(2,2,1,1)}$$

Theorem (Littlewood-Richardson, stated 1934, proven 1970's)

The coefficient $c_{\lambda\mu}^\nu$ is equal to the number of LR tableaux of shape λ/μ and type ν .



(LR tableaux of shape $(6,4,3)/(3,1)$ and type $(4,3,2)$. $c_{(3,1)(4,3,2)}^{(6,4,3)} = 2$)

Representation theoretic multiplicities II

Kronecker coefficients: $g(\lambda, \mu, \nu)$ – multiplicity of \mathbb{S}_λ in $\mathbb{S}_\nu \otimes \mathbb{S}_\mu$

$$\mathbb{S}_\nu \otimes \mathbb{S}_\mu = \bigoplus_{\lambda \vdash n} \mathbb{S}_\lambda^{\oplus g(\lambda, \mu, \nu)}$$

Representation theoretic multiplicities II

Kronecker coefficients: $g(\lambda, \mu, \nu)$ – multiplicity of \mathbb{S}_λ in $\mathbb{S}_\nu \otimes \mathbb{S}_\mu$

$$\mathbb{S}_\nu \otimes \mathbb{S}_\mu = \bigoplus_{\lambda \vdash n} \mathbb{S}_\lambda^{\oplus g(\lambda, \mu, \nu)}$$

$$s_\lambda(x \cdot y) = \sum_{\nu, \mu} g(\lambda, \mu, \nu) s_\nu(x) s_\mu(y)$$

$$g(\alpha, \beta, \gamma) = \sum_{\sigma \in S_a, \pi \in S_b, \rho \in S_c} \text{sgn}(\sigma) \text{sgn}(\pi) \text{sgn}(\rho) CT(\alpha + \sigma - \text{id}_a, \beta + \pi - \text{id}_b, \gamma + \rho - \text{id}_c).$$

$$(CT(\alpha, \beta, \gamma) = \#\{A \in \mathbb{Z}_{\geq 0}^{a \times b \times c} : \sum_{j,k} A_{i,j,k} = \alpha_i, \sum_{i,k} A_{i,j,k} = \beta_j, \sum_{i,j} A_{i,j,k} = \gamma_k\})$$

Representation theoretic multiplicities II

Kronecker coefficients: $g(\lambda, \mu, \nu)$ – multiplicity of \mathbb{S}_λ in $\mathbb{S}_\nu \otimes \mathbb{S}_\mu$

$$\mathbb{S}_\nu \otimes \mathbb{S}_\mu = \bigoplus_{\lambda \vdash n} \mathbb{S}_\lambda^{\oplus g(\lambda, \mu, \nu)}$$

$$s_\lambda(x \cdot y) = \sum_{\nu, \mu} g(\lambda, \mu, \nu) s_\nu(x) s_\mu(y)$$

$$g(\alpha, \beta, \gamma) = \sum_{\sigma \in S_a, \pi \in S_b, \rho \in S_c} \text{sgn}(\sigma) \text{sgn}(\pi) \text{sgn}(\rho) CT(\alpha + \sigma - \text{id}_a, \beta + \pi - \text{id}_b, \gamma + \rho - \text{id}_c).$$

$$(CT(\alpha, \beta, \gamma) = \#\{A \in \mathbb{Z}_{\geq 0}^{a \times b \times c} : \sum_{j,k} A_{i,j,k} = \alpha_i, \sum_{i,k} A_{i,j,k} = \beta_j, \sum_{i,j} A_{i,j,k} = \gamma_k\})$$

Plethysm coefficients: $GL_n \xrightarrow{\rho^\nu} GL_m \xrightarrow{\rho^\mu} GL_N: \rho^\mu \circ \rho^\nu : GL_n \rightarrow GL_N:$

$$\rho^\mu(\rho^\nu) = \bigoplus_{\lambda} V_\lambda^{\oplus a_\lambda(\mu[\nu])}$$

$a_\lambda(d[n])$ – multiplicity of V_λ in $\text{Sym}^d(\text{Sym}^n V)$ under GL action.

Representation theoretic multiplicities II

Kronecker coefficients: $g(\lambda, \mu, \nu)$ – multiplicity of \mathbb{S}_λ in $\mathbb{S}_\nu \otimes \mathbb{S}_\mu$

$$\mathbb{S}_\nu \otimes \mathbb{S}_\mu = \bigoplus_{\lambda \vdash n} \mathbb{S}_\lambda^{\oplus g(\lambda, \mu, \nu)}$$

$$s_\lambda(x \cdot y) = \sum_{\nu, \mu} g(\lambda, \mu, \nu) s_\nu(x) s_\mu(y)$$

$$g(\alpha, \beta, \gamma) = \sum_{\sigma \in S_a, \pi \in S_b, \rho \in S_c} \text{sgn}(\sigma) \text{sgn}(\pi) \text{sgn}(\rho) CT(\alpha + \sigma - \text{id}_a, \beta + \pi - \text{id}_b, \gamma + \rho - \text{id}_c).$$

$$(CT(\alpha, \beta, \gamma) = \#\{A \in \mathbb{Z}_{\geq 0}^{a \times b \times c} : \sum_{j,k} A_{i,j,k} = \alpha_i, \sum_{i,k} A_{i,j,k} = \beta_j, \sum_{i,j} A_{i,j,k} = \gamma_k\})$$

Plethysm coefficients: $GL_n \xrightarrow{\rho^\nu} GL_m \xrightarrow{\rho^\mu} GL_N: \rho^\mu \circ \rho^\nu : GL_n \rightarrow GL_N:$

$$\rho^\mu(\rho^\nu) = \bigoplus_{\lambda} V_\lambda^{\oplus a_\lambda(\mu[\nu])}$$

$a_\lambda(d[n])$ – multiplicity of V_λ in $\text{Sym}^d(\text{Sym}^n V)$ under GL action.

$$s_d[s_n(x)] = s_d(x_1^n, x_1^{n-1}x_2, \dots) = \sum_{\lambda} a_\lambda(d[n]) s_\lambda$$

Major problems in Algebraic Combinatorics

[Murnaghan, 1938]: $c_{\mu\nu}^{\lambda} = g((N - |\lambda|, \lambda), (N - |\mu|, \mu), (N - |\nu|, \nu))$ for $|\lambda| = |\mu| + |\nu|$ and N -large.

Major problems in Algebraic Combinatorics

[Murnaghan, 1938]: $c_{\mu\nu}^{\lambda} = g((N - |\lambda|, \lambda), (N - |\mu|, \mu), (N - |\nu|, \nu))$ for $|\lambda| = |\mu| + |\nu|$ and N -large.

Problem (Murnaghan 1938.. Lascoux, Garsia-Remmel 1980s... Stanley 2000)

Find a positive combinatorial interpretation for $g(\lambda, \mu, \nu)$, i.e. a family of combinatorial objects $\mathcal{O}_{\lambda, \mu, \nu}$, s.t. $g(\lambda, \mu, \nu) = \#\mathcal{O}_{\lambda, \mu, \nu}$.

Major problems in Algebraic Combinatorics

[Murnaghan, 1938]: $c_{\mu\nu}^{\lambda} = g((N - |\lambda|, \lambda), (N - |\mu|, \mu), (N - |\nu|, \nu))$ for $|\lambda| = |\mu| + |\nu|$ and N -large.

Problem (Murnaghan 1938.. Lascoux, Garsia-Remmel 1980s... Stanley 2000)

Find a positive combinatorial interpretation for $g(\lambda, \mu, \nu)$, i.e. a family of combinatorial objects $\mathcal{O}_{\lambda, \mu, \nu}$, s.t. $g(\lambda, \mu, \nu) = \#\mathcal{O}_{\lambda, \mu, \nu}$.

Combinatorial formulas for $g(\lambda, \mu, \nu)$:

- Two two-row partitions [Remmel–Whitehead, 1994; Blasiak–Mulmuley–Sohoni, 2015] ;
- One two-row and other restrictions [Ballantine-Orellana, 2006]
- One hook $\nu = (n - k, 1^k)$ [Blasiak 2012, Blasiak-Liu 2014]
- Other special cases [Bessenrodt-Bowman, Colmenarejo-Rosas, Garsia, Goupil, Ikenmeyer-Mulmuley-Walter, Pak-Panova, Tewari, Vallejo, Chenchen Zhao].

Major problems in Algebraic Combinatorics

[Murnaghan, 1938]: $c_{\mu\nu}^{\lambda} = g((N - |\lambda|, \lambda), (N - |\mu|, \mu), (N - |\nu|, \nu))$ for $|\lambda| = |\mu| + |\nu|$ and N -large.

Problem (Murnaghan 1938.. Lascoux, Garsia-Remmel 1980s... Stanley 2000)

Find a positive combinatorial interpretation for $g(\lambda, \mu, \nu)$, i.e. a family of combinatorial objects $\mathcal{O}_{\lambda, \mu, \nu}$, s.t. $g(\lambda, \mu, \nu) = \#\mathcal{O}_{\lambda, \mu, \nu}$.

Combinatorial formulas for $g(\lambda, \mu, \nu)$:

- Two two-row partitions [Remmel–Whitehead, 1994; Blasiak–Mulmuley–Sohoni, 2015] ;
- One two-row and other restrictions [Ballantine-Orellana, 2006]
- One hook $\nu = (n - k, 1^k)$ [Blasiak 2012, Blasiak-Liu 2014]
- Other special cases [Bessenrodt-Bowman, Colmenarejo-Rosas, Garsia, Goupil, Ikenmeyer-Mulmuley-Walter, Pak-Panova, Tewari, Vallejo, Chenchen Zhao].

Problem (Stanley 2000)

Find a positive combinatorial interpretation for $a_{\lambda}(\mu[\nu])$.

Major problems in Algebraic Combinatorics

[Murnaghan, 1938]: $c_{\mu\nu}^{\lambda} = g((N - |\lambda|, \lambda), (N - |\mu|, \mu), (N - |\nu|, \nu))$ for $|\lambda| = |\mu| + |\nu|$ and N -large.

Problem (Murnaghan 1938.. Lascoux, Garsia-Remmel 1980s... Stanley 2000)

Find a positive combinatorial interpretation for $g(\lambda, \mu, \nu)$, i.e. a family of combinatorial objects $\mathcal{O}_{\lambda, \mu, \nu}$, s.t. $g(\lambda, \mu, \nu) = \#\mathcal{O}_{\lambda, \mu, \nu}$.

Combinatorial formulas for $g(\lambda, \mu, \nu)$:

- Two two-row partitions [Remmel–Whitehead, 1994; Blasiak–Mulmuley–Sohoni, 2015] ;
- One two-row and other restrictions [Ballantine-Orellana, 2006]
- One hook $\nu = (n - k, 1^k)$ [Blasiak 2012, Blasiak-Liu 2014]
- Other special cases [Bessenrodt-Bowman, Colmenarejo-Rosas, Garsia, Goupil, Ikenmeyer-Mulmuley-Walter, Pak-Panova, Tewari, Vallejo, Chenchen Zhao].

Problem (Stanley 2000)

Find a positive combinatorial interpretation for $a_{\lambda}(\mu[\nu])$.

Applications beyond Combinatorics: **Geometric Complexity Theory** (VP vs VNP...), **Quantum Information Theory**, **Quantum Mechanics** (quantum marginal problem, orbital angular momentum) etc.

Major problems in Algebraic Combinatorics

[Murnaghan, 1938]: $c_{\mu\nu}^{\lambda} = g((N - |\lambda|, \lambda), (N - |\mu|, \mu), (N - |\nu|, \nu))$ for $|\lambda| = |\mu| + |\nu|$ and N -large.

Problem (Murnaghan 1938.. Lascoux, Garsia-Remmel 1980s... Stanley 2000)

Find a positive combinatorial interpretation for $g(\lambda, \mu, \nu)$, i.e. a family of combinatorial objects $\mathcal{O}_{\lambda, \mu, \nu}$, s.t. $g(\lambda, \mu, \nu) = \#\mathcal{O}_{\lambda, \mu, \nu}$.

Combinatorial formulas for $g(\lambda, \mu, \nu)$:

- Two two-row partitions [Remmel–Whitehead, 1994; Blasiak–Mulmuley–Sohoni, 2015] ;
- One two-row and other restrictions [Ballantine-Orellana, 2006]
- One hook $\nu = (n - k, 1^k)$ [Blasiak 2012, Blasiak-Liu 2014]
- Other special cases [Bessenrodt-Bowman, Colmenarejo-Rosas, Garsia, Goupil, Ikenmeyer-Mulmuley-Walter, Pak-Panova, Tewari, Vallejo, Chenchen Zhao].

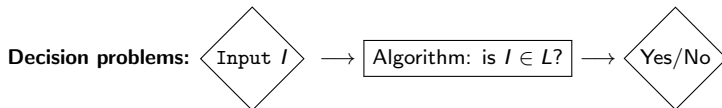
Problem (Stanley 2000)

Find a positive combinatorial interpretation for $a_{\lambda}(\mu[\nu])$.

Applications beyond Combinatorics: **Geometric Complexity Theory** (VP vs VNP...), **Quantum Information Theory, Quantum Mechanics** (quantum marginal problem, orbital angular momentum) etc.

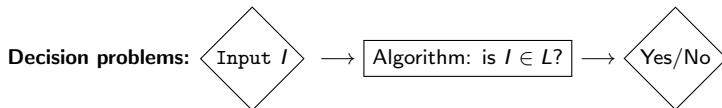
What is really a “combinatorial interpretation”?

Computational Complexity



Ex: $L = \text{Primes}$, I – an integer

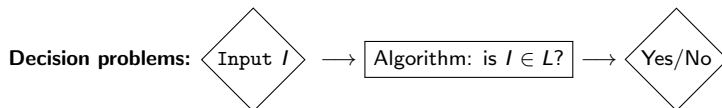
Computational Complexity



Ex: $L = \text{Primes}$, I – an integer

Input size(I) = n bits

Computational Complexity

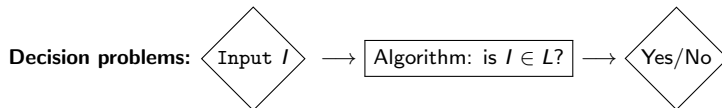


Ex: $L = \text{Primes}$, I – an integer

Input size(I) = n bits

P = yes/no answer in time $O(n^k)$ some fixed k .

Computational Complexity



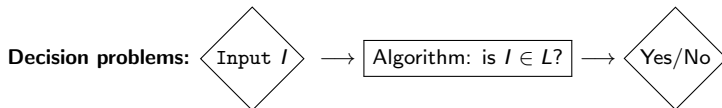
Ex: $L = \text{Primes}$, I – an integer

Input size(I) = n bits

P = yes/no answer in time $O(n^k)$ some fixed k .

Ex: PRIMES, Linear Programming, Graph connectivity ...

Computational Complexity



Ex: $L = \text{Primes}$, I – an integer

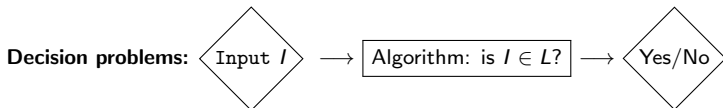
Input size(I) = n bits

P = yes/no answer in time $O(n^k)$ some fixed k .

Ex: PRIMES, Linear Programming, Graph connectivity ...

NP = “yes” can be *verified* in $O(n^k)$ for some fixed k , i.e. there is a “poly-time witness”.

Computational Complexity



Ex: $L = \text{Primes}$, I – an integer

Input size(I) = n bits

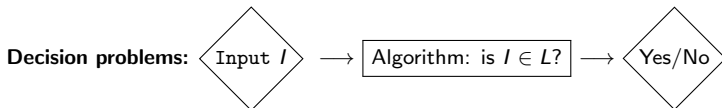
P = yes/no answer in time $O(n^k)$ some fixed k .

Ex: PRIMES, Linear Programming, Graph connectivity ...

NP = “yes” can be *verified* in $O(n^k)$ for some fixed k , i.e. there is a “poly-time witness”.

Ex: Input $I = G$ - graph, L =graphs with Hamiltonian cycles, the witness is the cycle $v_1 - \dots - v_m - v_1$

Computational Complexity



Ex: $L = \text{Primes}$, I – an integer

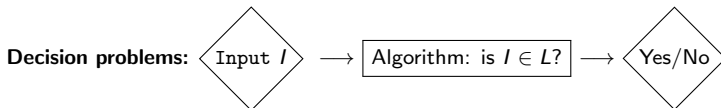
Input size(I) = n bits

P = yes/no answer in time $O(n^k)$ some fixed k .

NP = “yes” can be *verified* in $O(n^k)$ for some fixed k , i.e. there is a “poly-time witness”.

P vs NP Millennium problem: [Cook'71-Levin'73] Is $P \neq NP$?

Computational Complexity



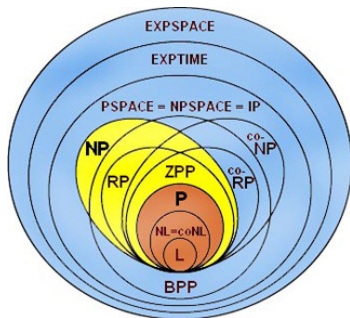
Ex: $L = \text{Primes}$, I – an integer

Input size(I) = n bits

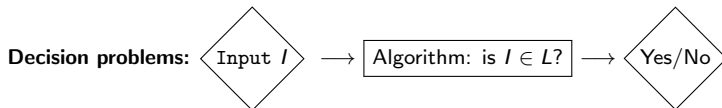
P = yes/no answer in time $O(n^k)$ some fixed k .

NP = “yes” can be verified in $O(n^k)$ for some fixed k , i.e. there is a “poly-time witness”.

P vs NP Millennium problem: [Cook'71-Levin'73] Is $P \neq NP$?



Computational Complexity

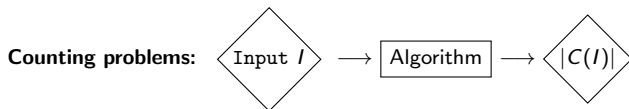


Ex: $L = \text{Primes}$, I – an integer

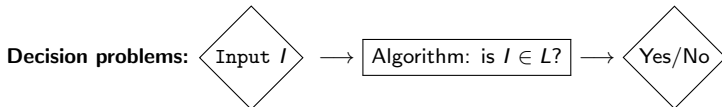
Input size(I) = n bits

P = yes/no answer in time $O(n^k)$ some fixed k .

NP = “yes” can be *verified* in $O(n^k)$ for some fixed k , i.e. there is a “poly-time witness”.



Computational Complexity

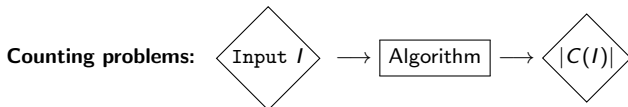


Ex: $L = \text{Primes}$, I – an integer

Input size(I) = n bits

P = yes/no answer in time $O(n^k)$ some fixed k .

NP = “yes” can be *verified* in $O(n^k)$ for some fixed k , i.e. there is a “poly-time witness”.



FP = answer in time $O(n^k)$ some fixed k .

Ex: Determinant, Spanning trees, recursions ...

#P = $\#\{y : \text{size}(y) \leq n^k, M(I, y) = 1\}$
for some fixed k and $M \in \text{P}$.

$$= \sum_{y \in \{0,1\}^{n^k}} M(I, y)$$

Ex: Input $I = G$ - graph, output – number of Hamiltonian cycles in G .

Combinatorial Interpretation via Computational Complexity

Counting and characterizing combinatorial objects given input data I

Solve: is $I \in L$, compute $|C(I)|$

Combinatorial Interpretation via Computational Complexity

Counting and characterizing combinatorial objects given input data I

Solve: is $I \in L$, compute $|C(I)|$

“Nice formula”

Product formulas, determinants etc

The problem is in P, FP

Combinatorial Interpretation via Computational Complexity

Counting and characterizing combinatorial objects given input data I

Solve: is $I \in L$, compute $|C(I)|$

“Nice formula”

Product formulas, determinants etc

The problem is in P, FP

Positive combinatorial interpretation

Ex: Littlewood-Richardson rule

The problem is in #P

Combinatorial Interpretation via Computational Complexity

Counting and characterizing combinatorial objects given input data I

Solve: is $I \in L$, compute $|C(I)|$

“Nice formula”

Product formulas, determinants etc

The problem is in P, FP

Positive combinatorial interpretation

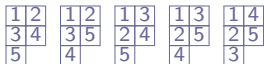
Ex: Littlewood-Richardson rule

The problem is in #P

No “combinatorial interpretation”
stay tuned

The problem is not in #P

Complexity of Computing SYTs and SSYT, warmup



Counting problem #SYT:

Input: Young diagram of λ

Output: Number of SYTs of shape λ ,
i.e. f^λ .

Complexity of Computing SYTs and SSYT, warmup



Counting problem #SYT:

Input: Young diagram of λ Output: Number of SYTs of shape λ ,
i.e. f^λ .

$$\dim \mathbb{S}_\lambda = f^\lambda = \#SYT(\lambda) = \frac{n!}{\prod_{(i,j) \in \lambda} (\lambda_i - i + \lambda'_j - j + 1)}.$$

Input $I = (\lambda_1, \lambda_2, \dots) = (\underbrace{11 \dots 1}_{\lambda_1}, \underbrace{1 \dots 1}_{\lambda_2}, \dots)$ is in *unary*, size = n .

Complexity of Computing SYTs and SSYT, warmup



Counting problem #SYT:

Input: Young diagram of λ Output: Number of SYTs of shape λ ,
i.e. f^λ .

$$\dim \mathbb{S}_\lambda = f^\lambda = \#SYT(\lambda) = \frac{n!}{\prod_{(i,j) \in \lambda} (\lambda_i - i + \lambda'_j - j + 1)}.$$

Input $I = (\lambda_1, \lambda_2, \dots) = (\underbrace{11 \dots 1}_{\lambda_1}, \underbrace{1 \dots 1}_{\lambda_2}, \dots)$ is in *unary*, size = n .

#SYT \in FP.

Complexity of Computing SYTs and SSYTs, warmup

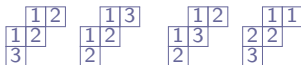


Counting problem #SYT:

Input: Young diagram of λ Output: Number of SYTs of shape λ ,
i.e. f^λ .

$$\dim \mathfrak{S}_\lambda = f^\lambda = \#SYT(\lambda) = \frac{n!}{\prod_{(i,j) \in \lambda} (\lambda_i - i + \lambda'_j - j + 1)}.$$

Input $I = (\lambda_1, \lambda_2, \dots) = (\underbrace{11 \dots 1}_{\lambda_1}, \underbrace{1 \dots 1}_{\lambda_2}, \dots)$ is in *unary*, size = n .

#SYT \in FP.

Counting problem #skewSSYT:

Input: Young diagrams of λ, μ Output: Number of SSYTs of shape λ/μ
and entries $\{1, \dots, m\}$.

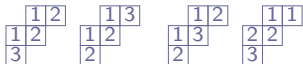
Complexity of Computing SYTs and SSYT, warmup



Counting problem #SYT:

Input: Young diagram of λ Output: Number of SYTs of shape λ ,
i.e. f^λ .

$$\dim \mathfrak{S}_\lambda = f^\lambda = \#\text{SYT}(\lambda) = \frac{n!}{\prod_{(i,j) \in \lambda} (\lambda_i - i + \lambda'_j - j + 1)}.$$

Input $I = (\lambda_1, \lambda_2, \dots) = (\underbrace{11 \dots 1}_{\lambda_1}, \underbrace{1 \dots 1}_{\lambda_2}, \dots)$ is in *unary*, size = n .#SYT \in FP.

Counting problem #skewSSYT:

Input: Young diagrams of λ, μ Output: Number of SSYT of shape λ/μ
and entries $\{1, \dots, m\}$.

$$= \det \left[\binom{m + \lambda_i - j - \mu_j + j - 1}{m - 1} \right]_{i,j=1}^{\ell(\lambda)}$$

#skewSSYT: \in FP

Complexity of Computing Multiplicities I

Littlewood-Richardson coefficients: $c_{\mu\nu}^{\lambda} = \text{mult}_{\lambda} V_{\mu} \otimes V_{\nu} = \#LR\text{-tableaux}$

$$c_{(4,3,2)(3,1)}^{(6,4,3)} = 2:$$



Complexity of Computing Multiplicities I

Littlewood-Richardson coefficients: $c_{\mu\nu}^{\lambda} = \text{mult}_{\lambda} V_{\mu} \otimes V_{\nu} = \#LR\text{-tableaux}$

$$c_{(4,3,2)(3,1)}^{(6,4,3)} = 2:$$



LR-Pos:

Input: λ, μ, ν

Output: Is $c_{\mu\nu}^{\lambda} > 0$?

ComputeLR:

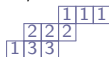
Input: λ, μ, ν

Output: Value of $c_{\mu\nu}^{\lambda}$.

Complexity of Computing Multiplicities I

Littlewood-Richardson coefficients: $c_{\mu\nu}^{\lambda} = \text{mult}_{\lambda} V_{\mu} \otimes V_{\nu} = \#LR\text{-tableaux}$

$$c_{(4,3,2)(3,1)}^{(6,4,3)} = 2:$$



LR-Pos:

Input: λ, μ, ν

Output: Is $c_{\mu\nu}^{\lambda} > 0$?

ComputeLR:

Input: λ, μ, ν

Output: Value of $c_{\mu\nu}^{\lambda}$.

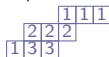
Theorem (cor. to Knutson-Tao'01)

LR-Pos is in P (even when the input is in binary).

Complexity of Computing Multiplicities I

Littlewood-Richardson coefficients: $c_{\mu\nu}^{\lambda} = \text{mult}_{\lambda} V_{\mu} \otimes V_{\nu} = \#LR\text{-tableaux}$

$$c_{(4,3,2)(3,1)}^{(6,4,3)} = 2:$$



LR-Pos:

Input: λ, μ, ν

Output: Is $c_{\mu\nu}^{\lambda} > 0$?

ComputeLR:

Input: λ, μ, ν

Output: Value of $c_{\mu\nu}^{\lambda}$.

Theorem (cor. to Knutson-Tao'01)

LR-Pos is in P (even when the input is in binary).

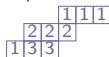
Theorem (Narayanan'05)

ComputeLR is #P-complete when the input is in binary (i.e. input size is $O(\ell(\lambda) \log(\lambda_1))$).

Complexity of Computing Multiplicities I

Littlewood-Richardson coefficients: $c_{\mu\nu}^{\lambda} = \text{mult}_{\lambda} V_{\mu} \otimes V_{\nu} = \#LR - \text{tableaux}$

$$c_{(4,3,2)(3,1)}^{(6,4,3)} = 2:$$



LR-Pos:

Input: λ, μ, ν

Output: Is $c_{\mu\nu}^{\lambda} > 0$?

ComputeLR:

Input: λ, μ, ν

Output: Value of $c_{\mu\nu}^{\lambda}$.

Theorem (cor. to Knutson-Tao'01)

LR-Pos is in P (even when the input is in binary).

Theorem (Narayanan'05)

ComputeLR is #P-complete when the input is in binary (i.e. input size is $O(\ell(\lambda) \log(\lambda_1))$).

Conjecture (Pak-Panova)

ComputeLR is strongly #P-complete, i.e. when input is in unary (input size is $O(n)$).
(Related to counting 2d contingency tables, and graphs with given degree sequence)

Complexity of Computing Multiplicities II

KronPos:

Input: λ, μ, ν

Output: Is $g(\lambda, \mu, \nu) > 0$?

ComputeKron:

Input: λ, μ, ν

Output: Value of $g(\lambda, \mu, \nu)$.

Conjecture (Mulmuley~2005)

$KronPos \in P$.

Complexity of Computing Multiplicities II

KronPos:

Input: λ, μ, ν

Output: Is $g(\lambda, \mu, \nu) > 0$?

ComputeKron:

Input: λ, μ, ν

Output: Value of $g(\lambda, \mu, \nu)$.

Conjecture (Mulmuley~2005)

KronPos \in P.

Theorem (Ikenmeyer-Mulmuley-Walter 2016)

KronPos is NP-hard.

Complexity of Computing Multiplicities II

KronPos:

Input: λ, μ, ν

Output: Is $g(\lambda, \mu, \nu) > 0$?

ComputeKron:

Input: λ, μ, ν

Output: Value of $g(\lambda, \mu, \nu)$.

Conjecture (Mulmuley~2005)

KronPos \in P.

Theorem (Ikenmeyer-Mulmuley-Walter 2016)

KronPos is NP-hard.

Conjecture (Mulmuley~2010)

KronPos \in NP and *ComputeKron* \in #P.

(Note that $\text{ComputeKron} \in \text{GapP}_{\geq 0} := \{f \in \#P - \#P, f \geq 0\}$)

Complexity of Computing Multiplicities II

KronPos:

Input: λ, μ, ν

Output: Is $g(\lambda, \mu, \nu) > 0$?

ComputeKron:

Input: λ, μ, ν

Output: Value of $g(\lambda, \mu, \nu)$.

Conjecture (Mulmuley~2005)

KronPos \in P.

Theorem (Ikenmeyer-Mulmuley-Walter 2016)

KronPos is NP-hard.

Conjecture (Mulmuley~2010)

KronPos \in NP and *ComputeKron* \in #P.

(Note that $\text{ComputeKron} \in \text{GapP}_{\geq 0} := \{f \in \#P - \#P, f \geq 0\}$)

Conjecture (Pak~2018)

ComputeKron is not in #P (if the polynomial hierarchy does not collapse), and so there would be no reasonable combinatorial interpretation.

Complexity of Computing Multiplicities II

KronPos:

Input: λ, μ, ν

Output: Is $g(\lambda, \mu, \nu) > 0$?

ComputeKron:

Input: λ, μ, ν

Output: Value of $g(\lambda, \mu, \nu)$.

Conjecture (Mulmuley~2005)

$KronPos \in P$.

Theorem (Ikenmeyer-Mulmuley-Walter 2016)

$KronPos$ is NP-hard.

Conjecture (Mulmuley~2010)

$KronPos \in NP$ and $ComputeKron \in \#P$.

(Note that $ComputeKron \in \text{GapP}_{\geq 0} := \{f \in \#P - \#P, f \geq 0\}$)

Conjecture (Pak~2018)

$ComputeKron$ is not in $\#P$ (if the polynomial hierarchy does not collapse), and so there would be no reasonable combinatorial interpretation.

PlethPos:

Input: λ, d, n

Output: Is $a_\lambda(d[n]) > 0$?

ComputePleth:

Input: λ, d, n

Output: Value of $a_\lambda(d[n])$.

Fischer-Ikenmeyer: PlethPos is NP-hard, ComputePleth is $\#P$ -hard.

Quantum algorithms for Kronecker coefficients

QMA – quantum analogue of NP

class of decision problems, where there exists a *quantum witness* that makes a poly-time quantum verifier accept with high probability.

$\text{NP} \subset \text{QMA}$

#BQP – quantum analogue of #P.

the dimension of the subspace of witnesses that a poly-time verifier accepts with high probability.

$\text{\#P} \subset \text{\#BQP}$.

Quantum algorithms for Kronecker coefficients

QMA – quantum analogue of NP

class of decision problems, where there exists a *quantum witness* that makes a poly-time quantum verifier accept with high probability.

$\text{NP} \subset \text{QMA}$

#BQP – quantum analogue of #P.

the dimension of the subspace of witnesses that a poly-time verifier accepts with high probability.

$\#P \subset \#BQP$.

Theorem (Bravyi-Chowdhury-Gosset-Havlicek-Zhu'23)

KronPos is in QMA. The problem of computing $f^\lambda f^\mu f^\nu g(\lambda, \mu, \nu)$ is in #BQP.

Theorem (Ikenmeyer-Subramanian'23)

ComputeKron is in #BQP.

Also stated by [Christandl-Harrow-Walter'15].

INTERESTING
ENGINEERING

NEWS INNOVATION ENTERTAINMENT VIDEOS MILITARY ENERGY SPACE

SCIENCE

Century-old math puzzle that stumped top supercomputers solved by quantum computing

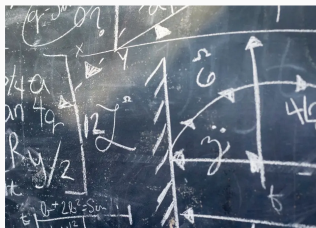
This study helps pinpoint where quantum computers can outperform classical computers.

Updated: Sep 11, 2025 07:35 AM EST



Aman Tripathi

3 days ago



This success highlights the real-world power of quantum advantage. (Representational image) [View image on Getty Images](#)

Researchers have successfully used a quantum algorithm to solve a complex century-old mathematical problem long considered impossible for even the most powerful conventional supercomputers.

The achievement has direct applications in fields including particle physics, material science, and data transmission.

"Is there a computational problem that has an efficient quantum algorithm but no efficient randomized algorithm? Quantum computing is driven by the belief that the answer is yes," said the researchers in a new study.



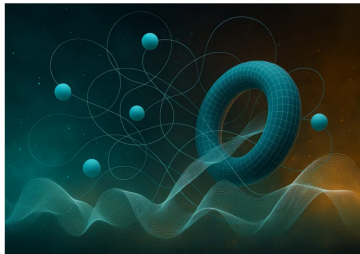
Los Alamos Reporter

The News from Los Alamos & Beyond

Home About The Los Alamos Reporter And Maire O'Neill Contact

SEPTEMBER 12, 2025

LANL: Quantum Computers Tackle A Century-Old Math Puzzle



A Los Alamos scientist has shown that quantum computers can be used to "factorize" group representations, a mathematical concept used in particle physics, engineering and many other fields. (Image created in GPT-4o)

LANL NEWS RELEASE

Los Alamos National Laboratory researcher Martín Larocca and Vojtěch Havlíček, a researcher at IBM, have shown that quantum computers can crack a family of complex problems involving group representations, which are sets of matrices that embody an abstract group, and that have stumped today's



Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $g(\lambda, \mu, \nu)$ in time $O\left(\frac{f^\mu f^\nu}{f^\lambda}\right)$.

Cor: if $f^\nu = \text{poly}(n)$, then there is a quantum poly-time algorithm for $g(\lambda, \mu, \nu)$

Question/conjecture[Larocca-Havlicek]: There is no classical poly-time algorithm for $g(\lambda, \mu, \nu)$ when f^ν is $\text{poly}(n)$?

Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $g(\lambda, \mu, \nu)$ in time $O\left(\frac{f^\mu f^\nu}{f^\lambda}\right)$.

Cor: if $f^\nu = \text{poly}(n)$, then there is a quantum poly-time algorithm for $g(\lambda, \mu, \nu)$

Question/conjecture[Larocca-Havlicek]: There is no classical poly-time algorithm for $g(\lambda, \mu, \nu)$ when f^ν is $\text{poly}(n)$?



Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $g(\lambda, \mu, \nu)$ in time $O\left(\frac{f^\mu f^\nu}{f^\lambda}\right)$.

Cor: if $f^\nu = \text{poly}(n)$, then there is a quantum poly-time algorithm for $g(\lambda, \mu, \nu)$

Question/conjecture[Larocca-Havlicek]: There is no classical poly-time algorithm for $g(\lambda, \mu, \nu)$ when f^ν is $\text{poly}(n)$?

Theorem (P'25)

Let $\lambda, \mu, \nu \vdash n$ and k be a constant, such that $f^\nu \leq n^k$. Then $g(\lambda, \mu, \nu)$ can be computed in time $O(n^{4k^2+2})$.

Cor: no quantum superpolynomial speedup in this case.

Proof sketch: Asymptotics: If $f^\nu \leq n^k$, then $n - \nu_1 \leq 4k^2$.

$$g(\lambda, \mu, \nu) = \sum_{\sigma \in S_{\ell(\nu)}} \text{sgn}(\sigma) \sum_{\alpha^i \vdash \nu_i + \sigma_i - i} c_{\alpha^1 \dots \alpha^\ell}^\lambda c_{\alpha^1 \dots \alpha^\ell}^\mu.$$

Quantum algorithms for plethysm coefficients

Theorem (Ikenmeyer-Subramanian'23)

ComputePleth $a_\lambda(d[n])$ is in $\#BQP$.

Note: general plethysms $a_\lambda(\mu[\nu])$ also (coming soon).

Quantum algorithms for plethysm coefficients

Theorem (Ikenmeyer-Subramanian'23)

ComputePleth $a_\lambda(d[n])$ is in #BQP.

Note: general plethysms $a_\lambda(\mu[\nu])$ also (coming soon).

Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $a^\lambda(\mu[\nu])$ *in time* $O\left(\frac{f^\lambda}{f^\mu(f^\nu)^{|\mu|}}\right)$.

Cor: if $f^\lambda = \text{poly}(n)$, then there is a quantum poly-time algorithm for $a_\lambda(d[m])$

Quantum algorithms for plethysm coefficients

Theorem (Ikenmeyer-Subramanian'23)

Compute $\text{Pleth } a_\lambda(d[n])$ is in $\#BQP$.

Note: general plethysms $a_\lambda(\mu[\nu])$ also (coming soon).

Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $a^\lambda(\mu[\nu])$ in time $O\left(\frac{f^\lambda}{f^\mu (f^\nu)^{|\mu|}}\right)$.

Cor: if $f^\lambda = \text{poly}(n)$, then there is a quantum poly-time algorithm for $a_\lambda(d[m])$

Theorem (P'25)

Let d, m be integers, $n = dm$ and $\lambda \vdash n$, such that $\lambda_1 \geq \ell(\lambda)$. Then the plethysm coefficient $a_{d,m}^\lambda$ can be computed in time

1. $O(n^{d\ell})$ where $\ell = \ell(\lambda)$.
2. $O(n^{4K^3(K+1)})$ where $f^\lambda \leq n^k$ and $K = 4k^2$ for arbitrary d, m .

In particular, we have a polynomial time algorithm for computing $a_{d,m}^\lambda$ if either d and $\ell(\lambda)$ are fixed, or d grows but the dimension f^λ grows at most polynomially.

Quantum algorithms for plethysm coefficients

Theorem (Ikenmeyer-Subramanian'23)

Compute $\text{Pleth } a_\lambda(d[n])$ is in $\#BQP$.

Note: general plethysms $a_\lambda(\mu[\nu])$ also (coming soon).

Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $a^\lambda(\mu[\nu])$ in time $O\left(\frac{f^\lambda}{f^{\mu(f^\nu)|\mu|}}\right)$.

Cor: if $f^\lambda = \text{poly}(n)$, then there is a quantum poly-time algorithm for $a_\lambda(d[m])$

Theorem (P'25)

Let d, m be integers, $n = dm$ and $\lambda \vdash n$, such that $\lambda_1 \geq \ell(\lambda)$. Then the plethysm coefficient $a_{d,m}^\lambda$ can be computed in time

1. $O(n^{d\ell})$ where $\ell = \ell(\lambda)$.
2. $O(n^{4K^3(K+1)})$ where $f^\lambda \leq n^k$ and $K = 4k^2$ for arbitrary d, m .

In particular, we have a polynomial time algorithm for computing $a_{d,m}^\lambda$ if either d and $\ell(\lambda)$ are fixed, or d grows but the dimension f^λ grows at most polynomially.

[Kahle-Michalek'15]: Poly-time algorithm when d, ℓ -fixed.

Cor 1: no quantum superpolynomial speedup in these cases.

Cor 2: $\text{COMPleth} \in \text{FP} \subset \#P$ for those families.

Quantum algorithms for plethysm coefficients

Theorem (Ikenmeyer-Subramanian'23)

Compute $\text{Pleth } a_\lambda(d[n])$ is in $\#BQP$.

Note: general plethysms $a_\lambda(\mu[\nu])$ also (coming soon).

Theorem (Larocca-Havlicek'24)

There exists a quantum algorithm computing $a^\lambda(\mu[\nu])$ in time $O\left(\frac{f^\lambda}{f^{\mu(f\nu)}|\mu|}\right)$.

Cor: if $f^\lambda = \text{poly}(n)$, then there is a quantum poly-time algorithm for $a_\lambda(d[m])$

Theorem (P'25)

Let d, m be integers, $n = dm$ and $\lambda \vdash n$, such that $\lambda_1 \geq \ell(\lambda)$. Then the plethysm coefficient $a_{d,m}^\lambda$ can be computed in time

1. $O(n^{d\ell})$ where $\ell = \ell(\lambda)$.
2. $O(n^{4K^3(K+1)})$ where $f^\lambda \leq n^k$ and $K = 4k^2$ for arbitrary d, m .

In particular, we have a polynomial time algorithm for computing $a_{d,m}^\lambda$ if either d and $\ell(\lambda)$ are fixed, or d grows but the dimension f^λ grows at most polynomially.

Cor 1: no quantum superpolynomial speedup in these cases.

Cor 2: $\text{COMP PLETH} \in \text{FP} \subset \#P$ for those families.

Proof sketch: counting points in polytopes Q :

$$a_{d,m}^\lambda = \sum_{\sigma \in S_{K+1}} \text{sgn}(\sigma) \sum_{r=1}^{4K^3+1} \sum_{(c_1, \dots, c_{r-1}) \in [1, 2K]^{r-1}} \sum_{\bar{j} \in [K+1]^{r-2}} |Q(\bar{j}, c, \hat{\lambda} + \delta(K) - \sigma(\delta))|$$

Characters of S_n

characters: $\text{char } \mathbb{S}_\lambda = \chi^\lambda : S_n \rightarrow \mathbb{C}$

$\chi^\lambda[\alpha] = \text{trace of the matrix in } \mathbb{S}_\lambda \text{ corresponding to a permutation of cycle type}$
 $\alpha = (\alpha_1, \alpha_2, \dots)$

Characters of S_n

characters: $\text{char } \mathbb{S}_\lambda = \chi^\lambda : S_n \rightarrow \mathbb{C}$

$\chi^\lambda[\alpha] = \text{trace of the matrix in } \mathbb{S}_\lambda \text{ corresponding to a permutation of cycle type } \alpha = (\alpha_1, \alpha_2, \dots)$

Murnaghan–Nakayama rule:

$$\chi^\lambda[\alpha] = \sum_{T : \text{MN tableaux, shape } \lambda, \text{ content } \alpha} (-1)^{ht(T)}$$

1	1	1	3	4	4
1	2	2	3	4	4
2	2	3	4		

— a M-N tableau T of shape $\lambda = (7, 6, 5)$,
content $\alpha = (4, 4, 5, 5)$,

$$ht(T) = (2 - 1) + (2 - 1) + (3 - 1) + (3 - 1) = 6.$$

Characters of S_n

characters: $\text{char } \mathbb{S}_\lambda = \chi^\lambda : S_n \rightarrow \mathbb{C}$

$\chi^\lambda[\alpha] = \text{trace of the matrix in } \mathbb{S}_\lambda \text{ corresponding to a permutation of cycle type } \alpha = (\alpha_1, \alpha_2, \dots)$

Murnaghan–Nakayama rule:

$$\chi^\lambda[\alpha] = \sum_{T : \text{MN tableaux, shape } \lambda, \text{ content } \alpha} (-1)^{ht(T)}$$

1	1	1	5	4	4
1	2	2	3	4	4
2	2	3	3	4	

— a M-N tableau T of shape $\lambda = (7, 6, 5)$,
content $\alpha = (4, 4, 5, 5)$,

$$ht(T) = (2 - 1) + (2 - 1) + (3 - 1) + (3 - 1) = 6.$$

Key players:

$$g(\lambda, \mu, \nu) = \frac{1}{n!} \sum_{w \in S_n} \chi^\lambda[w] \chi^\mu[w] \chi^\nu[w].$$

Characters of S_n

	id	$(1, 2)$	$(1, 2)(3, 4)$	$(1, 2, 3)$	$(1, 2, 3, 4)$
$\chi^{(4)}$	1	1	1	1	1
$\chi^{(1,1,1,1)}$	1	-1	1	1	-1
$\chi^{(3,1)}$	3	1	-1	0	-1
$\chi^{(2,1,1)}$	3	-1	-1	0	1
$\chi^{(2,2)}$	2	0	2	-1	0

Characters of S_n

	<i>id</i>	(1, 2)	(1, 2)(3, 4)	(1, 2, 3)	(1, 2, 3, 4)
$\chi^{(4)}$	1	1	1	1	1
$\chi^{(1,1,1,1)}$	1	-1	1	1	-1
$\chi^{(3,1)}$	3	1	-1	0	-1
$\chi^{(2,1,1)}$	3	-1	-1	0	1
$\chi^{(2,2)}$	2	0	2	-1	0

$$\sum_{\lambda \vdash n} \chi^\lambda(id)^2 = n!$$

Characters of S_n

	<i>id</i>	(1, 2)	(1, 2)(3, 4)	(1, 2, 3)	(1, 2, 3, 4)
$\chi^{(4)}$	1	1	1	1	1
$\chi^{(1,1,1,1)}$	1	-1	1	1	-1
$\chi^{(3,1)}$	3	1	-1	0	-1
$\chi^{(2,1,1)}$	3	-1	-1	0	1
$\chi^{(2,2)}$	2	0	2	-1	0

$$\sum_{\lambda \vdash n} \chi^\lambda(id)^2 = n!$$

$$\left(\begin{array}{|c|c|c|} \hline 1 & 2 & 4 \\ \hline 3 & & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & & \\ \hline \end{array} \right) \xleftrightarrow{RSK} 4123$$

Characters of S_n

	<i>id</i>	(1, 2)	(1, 2)(3, 4)	(1, 2, 3)	(1, 2, 3, 4)
$\chi^{(4)}$	1	1	1	1	1
$\chi^{(1,1,1,1)}$	1	-1	1	1	-1
$\chi^{(3,1)}$	3	1	-1	0	-1
$\chi^{(2,1,1)}$	3	-1	-1	0	1
$\chi^{(2,2)}$	2	0	2	-1	0

$$\sum_{\lambda \vdash n} \chi^\lambda(id)^2 = n!$$

$$\left(\begin{array}{|c|c|c|} \hline 1 & 2 & 4 \\ \hline 3 & & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & & \\ \hline \end{array} \right) \xleftrightarrow{RSK} 4123$$

$$\sum_{\lambda \vdash n} \chi^\lambda(w)^2 = \prod_i i^{c_i} c_i!$$

where c_i = number of cycles of length i in $w \in S_n$.

Characters of S_n

	<i>id</i>	(1, 2)	(1, 2)(3, 4)	(1, 2, 3)	(1, 2, 3, 4)
$\chi^{(4)}$	1	1	1	1	1
$\chi^{(1,1,1,1)}$	1	-1	1	1	-1
$\chi^{(3,1)}$	3	1	-1	0	-1
$\chi^{(2,1,1)}$	3	-1	-1	0	1
$\chi^{(2,2)}$	2	0	2	-1	0

$$\sum_{\lambda \vdash n} \chi^\lambda(id)^2 = n!$$

$$\left(\begin{array}{|c|c|c|} \hline 1 & 2 & 4 \\ \hline 3 & & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & & \\ \hline \end{array} \right) \xleftrightarrow{RSK} 4123$$

$$\sum_{\lambda \vdash n} \chi^\lambda(w)^2 = \prod_i i^{c_i} c_i!$$

where c_i = number of cycles of length i in $w \in S_n$.

COMPUTECHARSQ:

Input: $\lambda, \alpha \vdash n$, unary.

Output: the integer $\chi^\lambda(\alpha)^2$.

Characters of S_n

	<i>id</i>	(1, 2)	(1, 2)(3, 4)	(1, 2, 3)	(1, 2, 3, 4)
$\chi^{(4)}$	1	1	1	1	1
$\chi^{(1,1,1,1)}$	1	-1	1	1	-1
$\chi^{(3,1)}$	3	1	-1	0	-1
$\chi^{(2,1,1)}$	3	-1	-1	0	1
$\chi^{(2,2)}$	2	0	2	-1	0

$$\sum_{\lambda \vdash n} \chi^\lambda(id)^2 = n!$$

$$\left(\begin{array}{|c|c|c|} \hline 1 & 2 & 4 \\ \hline 3 & & \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & & \\ \hline \end{array} \right) \xleftrightarrow{RSK} 4123$$

$$\sum_{\lambda \vdash n} \chi^\lambda(w)^2 = \prod_i i^{c_i} c_i!$$

where c_i = number of cycles of length i in $w \in S_n$.

COMPUTECHARSQ:

Input: $\lambda, \alpha \vdash n$, unary.

Output: the integer $\chi^\lambda(\alpha)^2$.

Theorem (Ikenmeyer-Pak-P'22)

COMPUTECHARSQ $\notin \#P$ unless $PH = \Sigma_2^P$.

No nice combinatorial interpretation for $\chi^\lambda(\alpha)^2$

Set partitions

Ordered set partitions of items $\mathbf{a} = (a_1, \dots, a_m)$ into bins of sizes $\mathbf{b} = (b_1, \dots, b_k)$:

$$P(\mathbf{a}, \mathbf{b}) := \#\{(B_1, B_2, \dots, B_k) : B_1 \sqcup B_2 \sqcup \dots \sqcup B_k = [m], \sum_{i \in B_j} a_i = b_j \text{ for all } j = 1, \dots, k\}$$

$$P((\underbrace{1, 1, 1}_{4}, \underbrace{1, 1, 2}_{4}, \underbrace{2, 3}_{4}), (4, 4, 4)) = |\{(1+1+2, 1+3, 1+1+2), \dots\}| = 245$$

Set partitions

Ordered set partitions of items $\mathbf{a} = (a_1, \dots, a_m)$ into bins of sizes $\mathbf{b} = (b_1, \dots, b_k)$:

$$P(\mathbf{a}, \mathbf{b}) := \#\{(B_1, B_2, \dots, B_k) : B_1 \sqcup B_2 \sqcup \dots \sqcup B_k = [m], \sum_{i \in B_j} a_i = b_j \text{ for all } j = 1, \dots, k\}$$

$$P((\underbrace{1, 1, 1, 1}_{4}, \underbrace{1, 1, 2, 2}_{4}, 3), (\underbrace{4, 4, 4}_{4})) = |\{(1+1+2, 1+3, 1+1+2), \dots\}| = 245$$

Jacobi-Trudi/Frobenius character formula:

$$\chi^\lambda[\alpha] = \sum_{\sigma \in S_k} \text{sgn}(\sigma) P(\alpha, \lambda + \sigma - \text{id})$$

Set partitions

Ordered set partitions of items $\mathbf{a} = (a_1, \dots, a_m)$ into bins of sizes $\mathbf{b} = (b_1, \dots, b_k)$:

$$P(\mathbf{a}, \mathbf{b}) := \#\{(B_1, B_2, \dots, B_k) : B_1 \sqcup B_2 \sqcup \dots \sqcup B_k = [m], \sum_{i \in B_j} a_i = b_j \text{ for all } j = 1, \dots, k\}$$

$$P((\underbrace{1, 1, 1}_{4}, \underbrace{1, 1, 2}_{4}, \underbrace{2, 3}_{4}), (4, 4, 4)) = |\{(1+1+2, 1+3, 1+1+2), \dots\}| = 245$$

Jacobi-Trudi/Frobenius character formula:

$$\chi^\lambda[\alpha] = \sum_{\sigma \in S_k} \text{sgn}(\sigma) P(\alpha, \lambda + \sigma - \text{id})$$

Proposition (IPP)

Let \mathbf{c} and \mathbf{d} be two sequences of nonnegative integers, such that $|\mathbf{c}| = |\mathbf{d}| + 6$. Then there are partitions λ and α of size $O(\ell|\mathbf{c}|)$ determined in linear time, such that

$$\chi^\lambda(\alpha) = P(\mathbf{c}, \bar{\mathbf{d}}) - P(\mathbf{c}, \bar{\mathbf{d}}'),$$

where $\bar{\mathbf{d}} := (2, 4, d_1, d_2, \dots)$ and $\bar{\mathbf{d}}' := (1, 5, d_1, d_2, \dots)$.

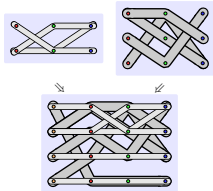
Matchings and $C=P$

Proposition (IPP)

For \forall two independent 3d matching problem instances E and E' , $\exists \mathbf{c}$ and \mathbf{d} , such that

$$\#3DM(E) - \#3DM(E') = \frac{1}{\delta} \left(P(\mathbf{c}, \mathbf{d}) - P(\mathbf{c}, \mathbf{d}') \right) = \frac{1}{\delta} \chi^\lambda(\alpha).$$

where δ is a fixed multiplicity factor, number of orderings.



¹A hypothesis widely believed to be false, similar to $P \neq NP$

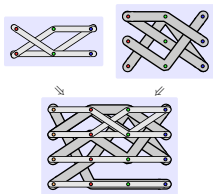
Matchings and $C=P$

Proposition (IPP)

For \forall two independent 3d matching problem instances E and E' , $\exists \mathbf{c}$ and \mathbf{d} , such that

$$\#3DM(E) - \#3DM(E') = \frac{1}{\delta} \left(P(\mathbf{c}, \bar{\mathbf{d}}) - P(\mathbf{c}, \bar{\mathbf{d}}') \right) = \frac{1}{\delta} \chi^\lambda(\alpha).$$

where δ is a fixed multiplicity factor, number of orderings.



$\Rightarrow [\chi = 0]$ is $C=P := [\underbrace{\text{GapP}}_{\#P - \#P} = 0]$ -complete.

¹A hypothesis widely believed to be false, similar to $P \neq NP$

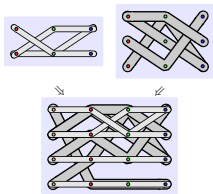
Matchings and $C=P$

Proposition (IPP)

For \forall two independent 3d matching problem instances E and E' , $\exists \mathbf{c}$ and \mathbf{d} , such that

$$\#3DM(E) - \#3DM(E') = \frac{1}{\delta} \left(P(\mathbf{c}, \bar{\mathbf{d}}) - P(\mathbf{c}, \bar{\mathbf{d}}') \right) = \frac{1}{\delta} \chi^\lambda(\alpha).$$

where δ is a fixed multiplicity factor, number of orderings.



$\Rightarrow [\chi = 0]$ is $C=P := [\underbrace{\text{GapP}}_{\#P - \#P} = 0]$ -complete.

If $\chi^2 \in \#P \Rightarrow [\chi^2 > 0] \in \text{NP}$, so $[\chi \neq 0] \in \text{NP}$
and hence $[\chi = 0] \in \text{coNP}$.

¹A hypothesis widely believed to be false, similar to $P \neq \text{NP}$

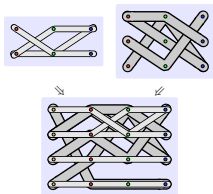
Matchings and $C=P$

Proposition (IPP)

For \forall two independent 3d matching problem instances E and E' , $\exists \mathbf{c}$ and \mathbf{d} , such that

$$\#3DM(E) - \#3DM(E') = \frac{1}{\delta} \left(P(\mathbf{c}, \bar{\mathbf{d}}) - P(\mathbf{c}, \bar{\mathbf{d}}') \right) = \frac{1}{\delta} \chi^\lambda(\alpha).$$

where δ is a fixed multiplicity factor, number of orderings.



$\Rightarrow [\chi = 0]$ is $C=P := [\underbrace{\text{GapP}}_{\#P - \#P} = 0]$ -complete.

If $\chi^2 \in \#P \Rightarrow [\chi^2 > 0] \in \text{NP}$, so $[\chi \neq 0] \in \text{NP}$
and hence $[\chi = 0] \in \text{coNP}$.

$\Rightarrow C=P \subset \text{coNP} \Rightarrow \text{since } \text{PH} \subset \text{NP}^{C=P} \text{ (Tarui'91)}$
then $\text{PH} \subset \text{NP}^{\text{coNP}}$, so $\text{PH} = \Sigma_2^P$

¹A hypothesis widely believed to be false, similar to $P \neq \text{NP}$

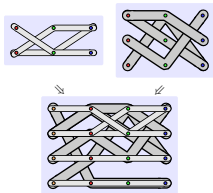
Matchings and $C=P$

Proposition (IPP)

For \forall two independent 3d matching problem instances E and E' , $\exists \mathbf{c}$ and \mathbf{d} , such that

$$\#3DM(E) - \#3DM(E') = \frac{1}{\delta} \left(P(\mathbf{c}, \mathbf{d}) - P(\mathbf{c}, \mathbf{d}') \right) = \frac{1}{\delta} \chi^\lambda(\alpha).$$

where δ is a fixed multiplicity factor, number of orderings.



$\Rightarrow [\chi = 0]$ is $C=P := [\underbrace{\text{GapP}}_{\#P - \#P} = 0]$ -complete.

If $\chi^2 \in \#P \Rightarrow [\chi^2 > 0] \in \text{NP}$, so $[\chi \neq 0] \in \text{NP}$
and hence $[\chi = 0] \in \text{coNP}$.

$\Rightarrow C=P \subset \text{coNP} \Rightarrow \text{since } \text{PH} \subset \text{NP}^{C=P} \text{ (Tarui'91)}$
then $\text{PH} \subset \text{NP}^{\text{coNP}}$, so $\text{PH} = \Sigma_2^P$

Theorem (Ikenmeyer-Pak-P'22)

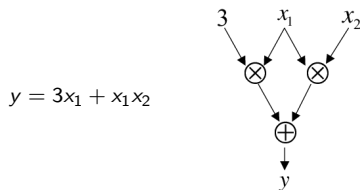
Let $\chi^2 : (\lambda, \pi) \mapsto (\chi^\lambda(\pi))^2$, where $\lambda \vdash n$ and $\pi \in S_n$. If $\chi^2 \in \#P$, then the polynomial hierarchy collapses to the second level: $\text{PH} = \Sigma_2^P = \text{NP}^1$.

No nice combinatorial interpretation for $\chi^\lambda(\alpha)^2$

¹A hypothesis widely believed to be false, similar to $P \neq \text{NP}$

Algebraic P vs NP: VP vs VNP

Arithmetic Circuits:



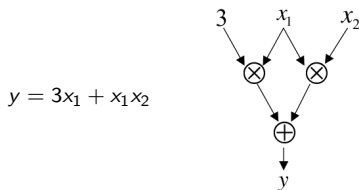
Input: X_1, \dots, X_n and constants from \mathbb{F} .

Circuit: nodes are $+$, $-$, \times , \div gates.

Output: Polynomial $y = f_n \in \mathbb{F}[X_1, \dots, X_n]$.

Algebraic P vs NP: VP vs VNP

Arithmetic Circuits:



Input: X_1, \dots, X_n and constants from \mathbb{F} .

Circuit: nodes are $+$, $-$, \times , \div gates.

Output: Polynomial $y = f_n \in \mathbb{F}[X_1, \dots, X_n]$.

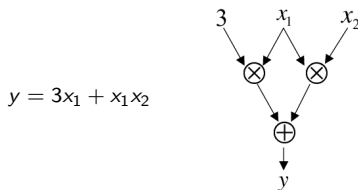
Class VP (Valliant's P): polynomials that can be computed with circuits with $\text{poly}(n)$ nodes

Class VNP (Valliant's NP): polynomials f_n , s.t. $\exists g_n \in \text{VP}$ with

$$f_n = \sum_{b \in \{0,1\}^n} g_n(X_1, \dots, X_n, b_1, \dots, b_n).$$

Algebraic P vs NP: VP vs VNP

Arithmetic Circuits:



Input: X_1, \dots, X_n and constants from \mathbb{F} .

Circuit: nodes are $+$, $-$, \times , \div gates.

Output: Polynomial $y = f_n \in \mathbb{F}[X_1, \dots, X_n]$.

Class VP (Valliant's P): polynomials that can be computed with circuits with $\text{poly}(n)$ nodes

Class VNP (Valliant's NP): polynomials f_n , s.t. $\exists g_n \in \text{VP}$ with

$$f_n = \sum_{b \in \{0,1\}^n} g_n(X_1, \dots, X_n, b_1, \dots, b_n).$$

Theorem[Bürgisser]:

If $\text{VP} = \text{VNP}$ over finite \mathbb{F} or over \mathbb{C} when GRH holds, then $\text{P} = \text{NP}$.

Geometric Complexity Theory in a Nutshell

VP vs VNP : permanent vs determinant

$$\det_n := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n x_{i, \sigma(i)} \quad \operatorname{per}_m := \sum_{\sigma \in S_m} \prod_{i=1}^m x_{i, \sigma(i)}$$

Geometric Complexity Theory in a Nutshell

VP vs VNP : permanent vs determinant

$$\det_n := \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n x_{i, \sigma(i)} \quad \text{per}_m := \sum_{\sigma \in S_m} \prod_{i=1}^m x_{i, \sigma(i)}$$

Theorem [Valiant'78]: det is VP -universal, per is VNP-universal.

$$\text{VP} \neq \text{VNP} \quad \text{if} \quad x_{11}^{n-m} \text{per}_m \neq \det_n[\mathbf{Ax}^T] \quad \text{for } n = \text{poly}(m). \quad 2$$

Geometric Complexity Theory in a Nutshell

VP vs VNP : permanent vs determinant

$$\det_n := \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n x_{i, \sigma(i)} \quad \text{per}_m := \sum_{\sigma \in S_m} \prod_{i=1}^m x_{i, \sigma(i)}$$

Theorem [Valiant'78]: det is VP -universal, per is VNP-universal.

$$\text{VP} \neq \text{VNP} \quad \text{if} \quad x_{11}^{n-m} \text{per}_m \neq \det_n[\mathbf{Ax}^T] \quad \text{for } n = \text{poly}(m). \quad 2$$

$$x_{11}^{n-m} \text{per}_m = \det_n[\mathbf{Ax}^T] \implies \overline{GL_{n^2} x_{11}^{n-m} \text{per}_m} \subset \overline{GL_{n^2} \det_n}$$

GCT program (Mulmuley and Sohoni): If so, show that $n > \text{poly}(m)$.

Geometric Complexity Theory in a Nutshell

VP vs VNP : permanent vs determinant

$$\det_n := \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n x_{i, \sigma(i)} \quad \text{per}_m := \sum_{\sigma \in S_m} \prod_{i=1}^m x_{i, \sigma(i)}$$

Theorem [Valiant'78]: det is VP -universal, per is VNP-universal.

$$\text{VP} \neq \text{VNP} \quad \text{if} \quad x_{11}^{n-m} \text{per}_m \neq \det_n[\mathbf{Ax}^T] \quad \text{for } n = \text{poly}(m). \quad 2$$

$$x_{11}^{n-m} \text{per}_m = \det_n[\mathbf{Ax}^T] \implies \overline{GL_{n^2} x_{11}^{n-m} \text{per}_m} \subset \overline{GL_{n^2} \det_n}$$

GCT program (Mulmuley and Sohoni): If so, show that $n > \text{poly}(m)$.

$$\mathbb{C}[\overline{GL_{n^2} \det_n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_{\lambda}^{\oplus \delta_{\lambda, d, n}}, \quad \mathbb{C}[\overline{GL_{n^2} \text{per}_m^n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_{\lambda}^{\oplus \gamma_{\lambda, d, n, m}},$$

Obstructions λ : if $\delta_{\lambda, d, n} < \gamma_{\lambda, d, n, m}$ for $n > \text{poly}(m) \implies \text{VP} \neq \text{VNP}$.

Geometric Complexity Theory in a Nutshell

VP vs VNP : permanent vs determinant

$$\det_n := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n x_{i, \sigma(i)} \quad \operatorname{per}_m := \sum_{\sigma \in S_m} \prod_{i=1}^m x_{i, \sigma(i)}$$

Theorem [Valiant'78]: \det is VP -universal, per is VNP-universal.

$$\text{VP} \neq \text{VNP} \quad \text{if} \quad x_{11}^{n-m} \operatorname{per}_m \neq \det_n[\mathbf{Ax}^T] \quad \text{for } n = \text{poly}(m). \quad 2$$

$$x_{11}^{n-m} \operatorname{per}_m = \det_n[\mathbf{Ax}^T] \implies \overline{GL_{n^2} x_{11}^{n-m} \operatorname{per}_m} \subset \overline{GL_{n^2} \det_n}$$

GCT program (Mulmuley and Sohoni): If so, show that $n > \text{poly}(m)$.

$$\mathbb{C}[\overline{GL_{n^2} \det_n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_\lambda^{\oplus \delta_{\lambda, d, n}}, \quad \mathbb{C}[\overline{GL_{n^2} \operatorname{per}_m^n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_\lambda^{\oplus \gamma_{\lambda, d, n, m}},$$

Obstructions λ : if $\delta_{\lambda, d, n} < \gamma_{\lambda, d, n, m}$ for $n > \text{poly}(m) \implies \text{VP} \neq \text{VNP}$.

Conjecture (Mulmuley and Sohoni)

There exist occurrence obstructions ($\delta_{\lambda, d, n} = 0 < \gamma_{\lambda, d, n, m}$) for $n > \text{poly}(m)$.

Geometric Complexity Theory in a Nutshell

VP vs VNP : permanent vs determinant

$$\det_n := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n x_{i, \sigma(i)} \quad \operatorname{per}_m := \sum_{\sigma \in S_m} \prod_{i=1}^m x_{i, \sigma(i)}$$

Theorem [Valiant'78]: \det is VP -universal, per is VNP-universal.

$$\text{VP} \neq \text{VNP} \quad \text{if} \quad x_{11}^{n-m} \operatorname{per}_m \neq \det_n[\mathbf{Ax}^T] \quad \text{for } n = \text{poly}(m). \quad 2$$

$$x_{11}^{n-m} \operatorname{per}_m = \det_n[\mathbf{Ax}^T] \implies \overline{GL_{n^2} x_{11}^{n-m} \operatorname{per}_m} \subset \overline{GL_{n^2} \det_n}$$

GCT program (Mulmuley and Sohoni): If so, show that $n > \text{poly}(m)$.

$$\mathbb{C}[\overline{GL_{n^2} \det_n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_\lambda^{\oplus \delta_{\lambda, d, n}}, \quad \mathbb{C}[\overline{GL_{n^2} \operatorname{per}_m^n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_\lambda^{\oplus \gamma_{\lambda, d, n, m}},$$

Obstructions λ : if $\delta_{\lambda, d, n} < \gamma_{\lambda, d, n, m}$ for $n > \text{poly}(m) \implies \text{VP} \neq \text{VNP}$.

Conjecture (Mulmuley and Sohoni)

There exist **occurrence obstructions** ($\delta_{\lambda, d, n} = 0 < \gamma_{\lambda, d, n, m}$) for $n > \text{poly}(m)$.

Theorem (Bürgisser-Ikenmeyer-P)

This Conjecture is false. There are no such occurrence obstructions for $n > m^{25}$.

Kronecker coefficients and GCT

$$\mathbb{C}[\overline{GL_{n^2} \det_n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_{\lambda}^{\oplus \delta_{\lambda, d, n}}, \quad \mathbb{C}[\overline{GL_{n^2} \text{per}_m^n}]_d \simeq \bigoplus_{\lambda \vdash nd} V_{\lambda}^{\oplus \gamma_{\lambda, d, n, m}},$$

Obstructions λ : if $\delta_{\lambda, d, n} < \gamma_{\lambda, d, n, m}$ for $n > \text{poly}(m) \implies \text{VP} \neq \text{VNP}$.

$$\delta_{\lambda, d, n} \leq g(\lambda, n^d, n^d) \quad \gamma_{\lambda, d, n, m} \leq a_{\lambda}(d[n])$$

Conjecture (GCT, Mulmuley and Sohoni)

There exist λ , s.t. $g(\lambda, n^d, n^d) = 0$ (so $\text{mult}_{\lambda} \mathbb{C}[\overline{GL_{n^2} \det_n}] = 0$) and $\gamma_{\lambda, d, n, m} > 0$ for some $n > \text{poly}(m)$.

Theorem (Ikenmeyer-P)

Let $n > 3m^4$, $\lambda \vdash nd$. If $g(\lambda, n^d, n^d) = 0$, then $\text{mult}_{\lambda}(\mathbb{C}[\overline{GL_{n^2} \text{per}_m^n}]) = 0$.

Theorem (Ikenmeyer-P)

If $\ell(\lambda) \leq m^2$, $\lambda_1 \geq nd - md$, $d > 3m^3$, and $n > 3m^4$, then $g(\lambda, n \times d, n \times d) > 0$, except for 6 special cases.

Theorem (Ikenmeyer-P)

For every partition ρ , let $n \geq |\rho|$, $d \geq 2$, $\lambda := (nd - |\rho|, \rho)$. Then $g(\lambda, n^d, n^d) \geq a_{\lambda}(d[n])$.

The End

Computing Kronecker and plethysm coefficients(?), S_n characters...and GCT...are...

The End

Computing Kronecker and plethysm coefficients(?), S_n characters...and GCT...are...



Thank you for your attention!