Plethysm is in #BQP

M. Christandl, A. W. Harrow, G. Panova, P. M. Posta, M. Walter

University of Copenhagen, QMATH

Computational Complexity Conference – CCC 2026

The General Problem: Branching Multiplicities

Consider two groups H and G with a homomorphism $\phi : H \to G$.

- Let $\rho : G \to \mathrm{GL}(W)$ be an irreducible representation (irrep) of G.
- We can define a representation of H by **restriction** (composition):

$$\pi = \rho \circ \phi : H \to \mathrm{GL}(W)$$

- Even if ρ is irreducible for G, its restriction π usually is **not** irreducible for H.

The General Problem: Branching Multiplicities

Consider two groups H and G with a homomorphism $\phi : H \to G$.

- Let $\rho : G \to \mathrm{GL}(W)$ be an irreducible representation (irrep) of G.
- We can define a representation of H by **restriction** (composition):

$$\pi = \rho \circ \phi : H \to \mathrm{GL}(W)$$

- Even if ρ is irreducible for G, its restriction π usually is **not** irreducible for H.

Branching Decomposition

The space W decomposes into a direct sum of H-irreps:

$$W \overset{H}{\cong} \bigoplus_{\lambda} \underbrace{V_\lambda \oplus V_\lambda \oplus \cdots \oplus V_\lambda}_{m_\lambda \text{ times}} \cong \bigoplus_{\lambda} V_\lambda \otimes \mathbb{C}^{m_\lambda}$$

The integers m_λ are the **branching multiplicities**.

Classic Examples of Multiplicities

Many interesting coefficients in algebraic combinatorics are specific branching multiplicities:

Classic Examples of Multiplicities

Many interesting coefficients in algebraic combinatorics are specific branching multiplicities:

1. **Littlewood-Richardson (LR) Coefficients** $c^{\lambda}_{\mu,\nu}$:
    - Branching multiplicity for $H = \mathrm{GL}(V) \to G = \mathrm{GL}(V) \times \mathrm{GL}(V)$.
    - Multiplicity of $\{\lambda\}_{\mathrm{GL}(V)}$ in the tensor product $\{\mu\}_{\mathrm{GL}(V)} \otimes \{\nu\}_{\mathrm{GL}(V)}$.

Classic Examples of Multiplicities

Many interesting coefficients in algebraic combinatorics are specific branching multiplicities:

1. **Littlewood-Richardson (LR) Coefficients** $c^{\lambda}_{\mu,\nu}$:
   - Branching multiplicity for $H = \mathrm{GL}(V) \to G = \mathrm{GL}(V) \times \mathrm{GL}(V)$.
   - Multiplicity of $\{\lambda\}_{\mathrm{GL}(V)}$ in the tensor product $\{\mu\}_{\mathrm{GL}(V)} \otimes \{\nu\}_{\mathrm{GL}(V)}$.

2. **Kronecker Coefficients** $g^{\lambda}_{\mu,\nu}$:
   - Branching multiplicity for $H = S_n \to G = S_n \times S_n$.
   - Multiplicity of $[\lambda]_{S_n}$ in the tensor product $[\mu]_{S_n} \otimes [\nu]_{S_n}$.

Classic Examples of Multiplicities

Many interesting coefficients in algebraic combinatorics are specific branching multiplicities:

1. **Littlewood-Richardson (LR) Coefficients** $c^{\lambda}_{\mu,\nu}$:
   - Branching multiplicity for $H = \mathrm{GL}(V) \to G = \mathrm{GL}(V) \times \mathrm{GL}(V)$.
   - Multiplicity of $\{\lambda\}_{\mathrm{GL}(V)}$ in the tensor product $\{\mu\}_{\mathrm{GL}(V)} \otimes \{\nu\}_{\mathrm{GL}(V)}$.
2. **Kronecker Coefficients** $g^{\lambda}_{\mu,\nu}$:
   - Branching multiplicity for $H = S_n \to G = S_n \times S_n$.
   - Multiplicity of $[\lambda]_{S_n}$ in the tensor product $[\mu]_{S_n} \otimes [\nu]_{S_n}$.
3. **Plethysm Coefficients** $a^{\lambda}_{\mu,\nu}$:
   - Branching multiplicity for $H = \mathrm{GL}(V) \to G = \mathrm{GL}(\{\nu\}_H)$.
   - Irreps can be viewed as mappings: $\rho_\nu : \mathrm{GL}(V) \to \mathrm{GL}(\{\nu\}_{\mathrm{GL}(V)})$.
   - The plethysm is s composition with an irrep for G: $\rho_\mu \circ \rho_\nu : \mathrm{GL}(V) \to \mathrm{GL}(\{\mu\}_G)$.
   - $a^{\lambda}_{\mu,\nu}$ is the multiplicity of $\{\lambda\}_{\mathrm{GL}(V)}$ in this composed mapping.

Classical Complexity Landscape

How hard is it to compute these m_λ, or even decide if $m_\lambda > 0$?

Coefficient	Positivity ($m_\lambda > 0$)	Computation (m_λ)
LR	In P (Saturation Thm)	In #P [1]
Kronecker	NP-hard	#P-hard, in GapP
Plethysm	NP-hard	#P-hard, in GapP

[1] complete for binary encoding

Classical Complexity Landscape

How hard is it to compute these m_λ, or even decide if $m_\lambda > 0$?

Coefficient	Positivity ($m_\lambda > 0$)	Computation (m_λ)
LR	In P (Saturation Thm)	In #P [1]
Kronecker	NP-hard	#P-hard, in GapP
Plethysm	NP-hard	#P-hard, in GapP

An open problem at the intersection of complexity and combinatorics

Are Kronecker and Plethysm coefficients in #P?

Finding a **positive combinatorial interpretation** (showing they count easily verifiable combinatorial objects) is a major open problem (Stanley's Positivity Problems 9 and 10).

[1] complete for binary encoding

Classical Complexity Landscape

How hard is it to compute these m_λ, or even decide if $m_\lambda > 0$?

Coefficient	Positivity ($m_\lambda > 0$)	Computation (m_λ)
LR	In P (Saturation Thm)	In #P [1]
Kronecker	NP-hard, QMA[2]	#P-hard, in GapP, in #BQP[2]
Plethysm	NP-hard	#P-hard, in GapP

An open problem at the intersection of complexity and combinatorics

Are Kronecker and Plethysm coefficients in #P?

Finding a **positive combinatorial interpretation** (showing they count easily verifiable combinatorial objects) is a major open problem (Stanley's Positivity Problems 9 and 10).

[1] complete for binary encoding

[2] Partial proof in [Bravyi, Chowdhury, Gosset, Havlicek, Zhu, 2024], completed and extended in [Ikenmeyer, Subramanian, 2025]

How hard is it to compute these m_λ, or even decide if $m_\lambda > 0$?

Coefficient	Positivity ($m_\lambda > 0$)	Computation (m_λ)
LR	In P (Saturation Thm)	In #P [1]
Kronecker	NP-hard, QMA[2]	#P-hard, in GapP, in #BQP[2]
Plethysm	NP-hard, QMA	#P-hard, in GapP, in #BQP

An open problem at the intersection of complexity and combinatorics

Are Kronecker and Plethysm coefficients in #P?

Finding a **positive combinatorial interpretation** (showing they count easily verifiable combinatorial objects) is a major open problem (Stanley's Positivity Problems 9 and 10).

[1] complete for binary encoding

[2] Partial proof in [Bravyi, Chowdhury, Gosset, Havlicek, Zhu, 2024], completed and extended in [Ikenmeyer, Subramanian, 2025]

What is #BQP?

#BQP is the quantum computing analogue of #P.

What is #BQP?

#BQP is the quantum computing analogue of #P.

Definition (Quantum Witness Counting)

A function $f(x)$ is in #BQP if for any x there exists a uniform polynomial-time quantum circuit with witness space $W = W_{\text{good}} \oplus W_{\text{bad}}$ such that

- for any state $|\psi\rangle \in W_{\text{good}}$, the probability of accepting is $> 2/3$,
- for any state $|\phi\rangle \in W_{\text{bad}}$, the probability of accepting is $< 1/3$,

and the function evaluates to the dimension of the good subspace: $f(x) = \dim(W_{\text{good}})$.

What is #BQP?

#BQP is the quantum computing analogue of #P.

Definition (Quantum Witness Counting)

A function $f(x)$ is in #BQP if for any x there exists a uniform polynomial-time quantum circuit with witness space $W = W_{\text{good}} \oplus W_{\text{bad}}$ such that

- for any state $|\psi\rangle \in W_{\text{good}}$, the probability of accepting is $> 2/3$,
- for any state $|\phi\rangle \in W_{\text{bad}}$, the probability of accepting is $< 1/3$,

and the function evaluates to the dimension of the good subspace: $f(x) = \dim(W_{\text{good}})$.

How do branching multiplicities fit in this? By definition branching multiplicities count the dimension of a multiplicity space. If we can build an efficient quantum circuit that accepts states in this multiplicity space and rejects the rest, we immediately place the coefficient in #BQP.

Bridging Groups: Schur-Weyl Duality

Can we change which group to compute with?

Schur-Weyl Duality

The tensor power space $V^{\otimes n}$ has natural, commuting actions of both the general linear group $\mathrm{GL}(V)$ and the symmetric group S_n. It decomposes as:

$$V^{\otimes n} \cong \bigoplus_{\lambda \vdash n} \{\lambda\}_{\mathrm{GL}(V)} \otimes [\lambda]_{S_n}$$

Bridging Groups: Schur-Weyl Duality

Can we change which group to compute with?

Schur-Weyl Duality

The tensor power space $V^{\otimes n}$ has natural, commuting actions of both the general linear group $\mathrm{GL}(V)$ and the symmetric group S_n. It decomposes as:

$$V^{\otimes n} \cong \bigoplus_{\lambda \vdash n} \{\lambda\}_{\mathrm{GL}(V)} \otimes [\lambda]_{S_n}$$

- **Consequence:** Multiplicities defined for the symmetric group S_n (like Kronecker coefficients) can be reformulated using the representation theory of $\mathrm{GL}(V)$.
- The reverse is also true (LR and plethysm can be stated via S_n).
- This allows us to pick the most algorithmically convenient algebraic setting.

Prior Methods vs. Our Method

Prior Approaches:

- Interpret all coefficients on the level of the symmetric group S_n.
- Use **generalized phase estimation** and a **Quantum Fourier Transform** over S_n.

Prior Methods vs. Our Method

Prior Approaches:

- Interpret all coefficients on the level of the symmetric group S_n.
- Use **generalized phase estimation** and a **Quantum Fourier Transform** over S_n.

Our Approach:

- We interpret the coefficients on the level of the general linear group $\mathrm{GL}(V)$.
- We replace the quantum Fourier transform over S_n with a **Quantum Schur Transform**.
- This framework captures previously studied coefficients and can handle the general plethysm problem in polynomial time.

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$W = \{\mu\}_{\mathrm{GL}(\{\nu\}_H)}$$

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \stackrel{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$W = \quad \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m}$$

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$\begin{aligned} W = \quad & \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m} \\ \xrightarrow{\text{inv. Schur}} \quad & (\{\nu\}_H)^{\otimes m} \end{aligned}$$

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$\begin{aligned} W = \quad & \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m} \\ \xrightarrow{\text{inv. Schur}} \quad & (\{\nu\}_H)^{\otimes m} \xrightarrow{\text{add ancillas}} (\{\nu\}_H \otimes [\nu]_{S_n})^{\otimes m} \end{aligned}$$

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$\begin{aligned} W = \quad & \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m} \\ \xrightarrow{\text{inv. Schur}} \quad & (\{\nu\}_H)^{\otimes m} \xrightarrow{\text{add ancillas}} (\{\nu\}_H \otimes [\nu]_{S_n})^{\otimes m} \\ \xrightarrow{\text{inv. Schur}} \quad & (V^{\otimes n})^{\otimes m} \end{aligned}$$

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$\begin{aligned}
W = \quad & \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m} \\
\xrightarrow{\text{inv. Schur}} \quad & (\{\nu\}_H)^{\otimes m} \xrightarrow{\text{add ancillas}} (\{\nu\}_H \otimes [\nu]_{S_n})^{\otimes m} \\
\xrightarrow{\text{inv. Schur}} \quad & (V^{\otimes n})^{\otimes m} \qquad \xrightarrow{\text{forw. Schur}} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes [\lambda]_{S_{nm}}
\end{aligned}$$

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^{\lambda}_{\mu,\nu}}$$

To compute the plethysm coefficient $a^{\lambda}_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

$$\begin{aligned} W = \quad & \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m} \\ \xrightarrow{\text{inv. Schur}} \quad & (\{\nu\}_H)^{\otimes m} \xrightarrow{\text{add ancillas}} (\{\nu\}_H \otimes [\nu]_{S_n})^{\otimes m} \\ \xrightarrow{\text{inv. Schur}} \quad & (V^{\otimes n})^{\otimes m} \xrightarrow{\text{forw. Schur}} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes [\lambda]_{S_{nm}} \end{aligned}$$

1. **Embedding:** Add ancilla states and use *inverse* Schur Transforms to embed the witness as a state in the computational basis of $V^{\otimes nm}$.

The Strategy: Embed and Project

Let $\mu \vdash m$, $\nu \vdash n$, recall:

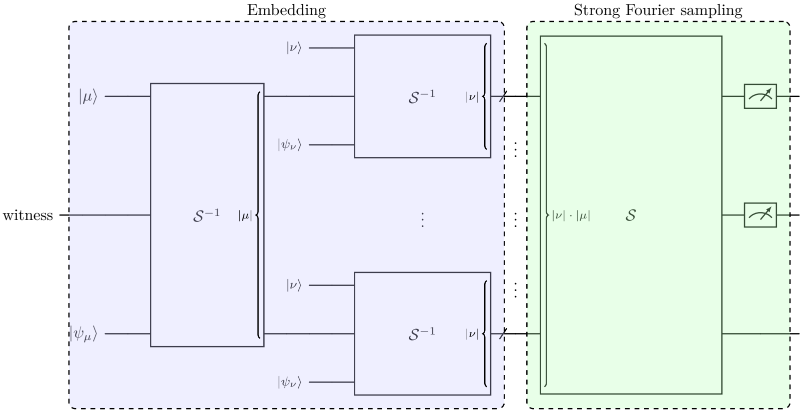
$$\{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \overset{H}{\cong} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes \mathbb{C}^{a^\lambda_{\mu,\nu}}$$

To compute the plethysm coefficient $a^\lambda_{\mu,\nu}$, we embed the abstract representation into a larger tensor space:

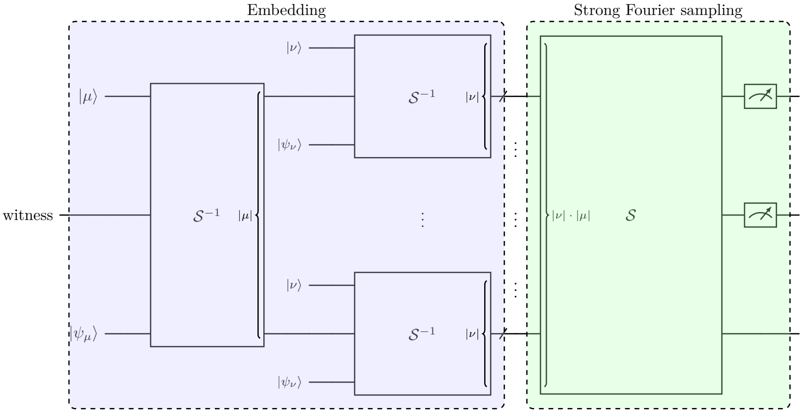
$$\begin{aligned}
W = \quad & \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \xrightarrow{\text{add ancilla}} \{\mu\}_{\mathrm{GL}(\{\nu\}_H)} \otimes [\mu]_{S_m} \\
\xrightarrow{\text{inv. Schur}} \quad & (\{\nu\}_H)^{\otimes m} \xrightarrow{\text{add ancillas}} (\{\nu\}_H \otimes [\nu]_{S_n})^{\otimes m} \\
\xrightarrow{\text{inv. Schur}} \quad & (V^{\otimes n})^{\otimes m} \xrightarrow{\text{forw. Schur}} \bigoplus_{\lambda \vdash nm} \{\lambda\}_H \otimes [\lambda]_{S_{nm}}
\end{aligned}$$

1. **Embedding:** Add ancilla states and use *inverse* Schur Transforms to embed the witness as a state in the computational basis of $V^{\otimes nm}$.
2. **Measurement:** Use a *forward* Schur Transform and measure. We accept if we obtain the label λ and a fixed state of the Weyl module.

The Quantum Circuit for Plethysm



The Quantum Circuit for Plethysm



As the chain of inclusions is H-equivariant, the operations act as the identity on the Weyl module. The total circuit implements a projective measurement onto the multiplicity subspace.

The High-Dimensional Schur Transform

The Challenge for Plethysm

The local dimension $d = \dim(\{\nu\}_H)$ can be **exponential** in the input size.

The standard Quantum Schur Transform algorithm on $(\mathbb{C}^d)^{\otimes N}$, scales as $\mathrm{poly}(d, N)$. Using this would destroy the polynomial-time requirement.

The High-Dimensional Schur Transform

The Challenge for Plethysm

The local dimension $d = \dim(\{\nu\}_H)$ can be **exponential** in the input size.

The standard Quantum Schur Transform algorithm on $(\mathbb{C}^d)^{\otimes N}$, scales as $\mathrm{poly}(d, N)$. Using this would destroy the polynomial-time requirement.

The Solution:
We utilize the recent **High-Dimensional Schur Transform**[3]. This algorithm's complexity scales as $\mathrm{poly}(\log d, N)$.

[3] [Burchardt, Fei, Grinko, Larocca, Ozols, Timmerman, Visnevskyi, 2025]

Summary of Results

- **Main Result:** The general plethysm problem is in #BQP.

Summary of Results

- **Main Result:** The general plethysm problem is in #BQP.
- **Generalization:** A generalized framework to prove membership in #BQP for all branching multiplicities for products of General Linear Groups. (Captures LR, Kronecker, Restriction, Kostka, etc.)

Summary of Results

- **Main Result:** The general plethysm problem is in #BQP.
- **Generalization:** A generalized framework to prove membership in #BQP for all branching multiplicities for products of General Linear Groups. (Captures LR, Kronecker, Restriction, Kostka, etc.)

Further directions

- Better understand the class #BQP and its relation to #P and GapP
- #BPP?
- Get quantum algorithms to actually compute these quantities (partial results in [Larocca, Havlicek, 2025])