Constant-Depth Circuits for Polynomial GCD over Any Characteristic

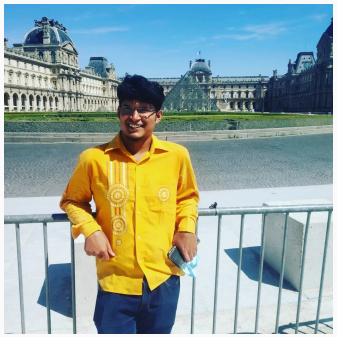
Understanding the complexity of symmetric polynomials

Varun Ramanathan

August 5, 2026

BARC, University of Copenhagen

Joint work with



Somnath Bhattacharjee (UofT)

Mrinal Kumar (TIFR)

Shanthanu S. Rai (TIFR)

Ramprasad Saptharishi (TIFR)

Shubhangi Saraf (UofT)

Polynomial GCD

$$f(x) = \prod_{i \in [d]} (x - \alpha_i) = x^d + f_{d-1} x^{d-1} + \ldots f_0 \in \mathbb{F}[x]$$

$$g(x) = \prod_{i \in [d]} (x - \beta_i) = x^d + g_{d-1} x^{d-1} + \ldots g_0 \in \mathbb{F}[x]$$

$\gcd(f, g)$ is the highest degree polynomial that divides both f and g.

Algorithmic question: given coefficients of f and g, compute the coefficients of $\gcd(f, g)$.

Polynomial GCD

Euclid

?

Parallel = circuits

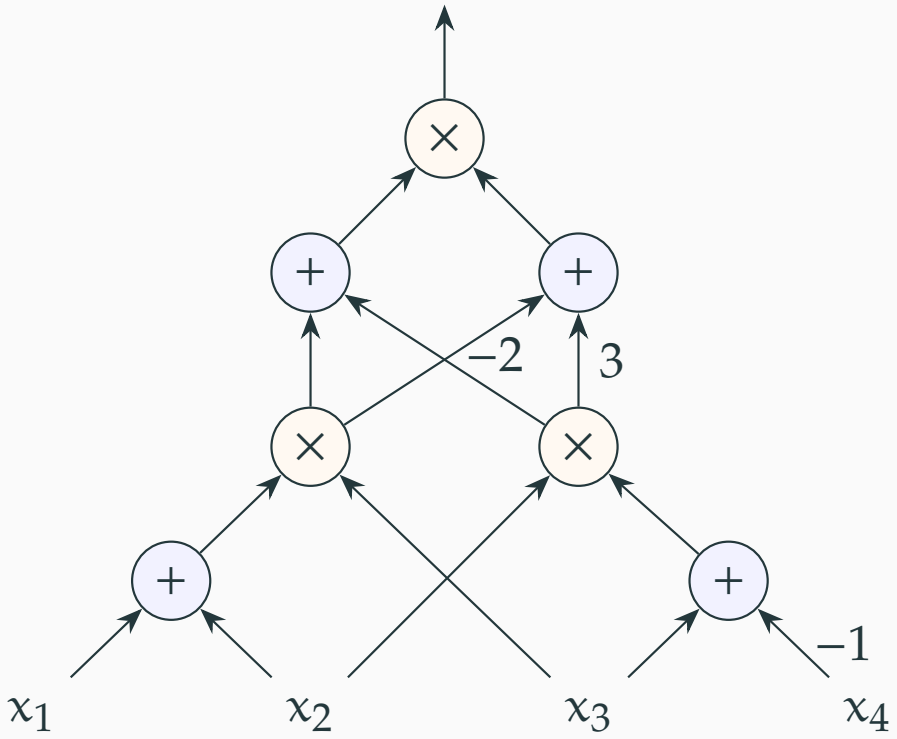
Boolean functions: shallow Boolean circuits.

Parallel = circuits

Boolean functions: shallow Boolean circuits.

Polynomials: shallow arithmetic circuits.

$$x_1^2x_3^2 + 3x_1x_3^2x_2 - x_1x_3x_2x_4 - 4x_2^2x_3^2 + 11x_2^2x_3x_4 - 6x_2^2x_4^2$$



Technicality

$\gcd$ is not continuous.

Example.

$$\gcd(x-\alpha, x-\beta) = \begin{cases} 1, & \text{if } \alpha \neq \beta, \\ x-\alpha, & \text{if } \alpha = \beta. \end{cases}$$

To avoid this issue, we will focus on the Resultant.

Resultant

$$f(x) = (x - \alpha_1)(x - \alpha_2)\dots(x - \alpha_d)$$

$$g(x) = (x - \beta_1)(x - \beta_2)\dots(x - \beta_d)$$

Resultant

$$f(x) = (x - \alpha_1)(x - \alpha_2)\dots(x - \alpha_d)$$

$$g(x) = (x - \beta_1)(x - \beta_2)\dots(x - \beta_d)$$

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

Resultant

$$f(x) = (x - \alpha_1)(x - \alpha_2)\dots(x - \alpha_d)$$

$$g(x) = (x - \beta_1)(x - \beta_2)\dots(x - \beta_d)$$

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

Fact: $\mathrm{Res}(f, g) = 0 \iff \gcd(f, g)$ is non-trivial.

Resultant

$$f(x) = (x - \alpha_1)(x - \alpha_2)\dots(x - \alpha_d)$$

$$g(x) = (x - \beta_1)(x - \beta_2)\dots(x - \beta_d)$$

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

Fact: $\mathrm{Res}(f, g) = 0 \iff \gcd(f, g)$ is non-trivial.

Given the coefficients of f and g, how easy is the Resultant?

Resultant is a polynomial

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

Fact: $\mathrm{Res}(f, g) \in \mathbb{F}[f_0, \ldots, f_{d-1}, g_0, \ldots, g_{d-1}]$

$$\mathrm{Res}(f, g) = \det(\mathrm{Syl}(f, g))$$

Complexity of the Resultant

$\mathrm{Res}(f, g)$: size $\mathrm{poly}(d)$, depth $O(\log^2 d)$ [Csanky 1976, Berkowitz 1983, . . .].

Complexity of the Resultant

$\mathrm{Res}(f, g)$: size $\mathrm{poly}(d)$, depth $O(\log^2 d)$ [Csanky 1976, Berkowitz 1983, . . .].

Complexity of the Resultant

$\mathrm{Res}(f, g)$: size $\mathrm{poly}(d)$, depth $O(\log^2 d)$ [Csanky 1976, Berkowitz 1983, . . .].

Theorem (Andrews and Wigderson 2024)

Over fields with char$(\mathbb{F})$ *zero or large,* $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Also for $\gcd$ and many other problems.

Our results

Theorem (BKRRSS)

Over any large enough field, $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Also for $\gcd$.

Resultant is a polynomial

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

$$\mathrm{Res}(f, g) \in \mathbb{F}[f_0, \ldots, f_{d-1}, g_0, \ldots, g_{d-1}]$$

Resultant is a polynomial

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

$$\mathrm{Res}(f, g) \in \mathbb{F}[f_0, \ldots, f_{d-1}, g_0, \ldots, g_{d-1}]$$

Reason: $\mathrm{Res}(f, g)$ does not change if you permute the roots of f, or if you permute the roots of g.

Resultant is a polynomial

$$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$$

$$\mathrm{Res}(f, g) \in \mathbb{F}[f_0, \ldots, f_{d-1}, g_0, \ldots, g_{d-1}]$$

Reason: $\mathrm{Res}(f, g)$ does not change if you permute the roots of f, or if you permute the roots of g.

Symmetric polynomials

$f(z_1, \ldots, z_n)$ is a symmetric polynomial if for any permutation $\pi : [n] \to [n]$,

$$f(z_{\pi(1)}, \ldots, z_{\pi(n)}) = f(z_1, \ldots, z_n).$$

Symmetric polynomials

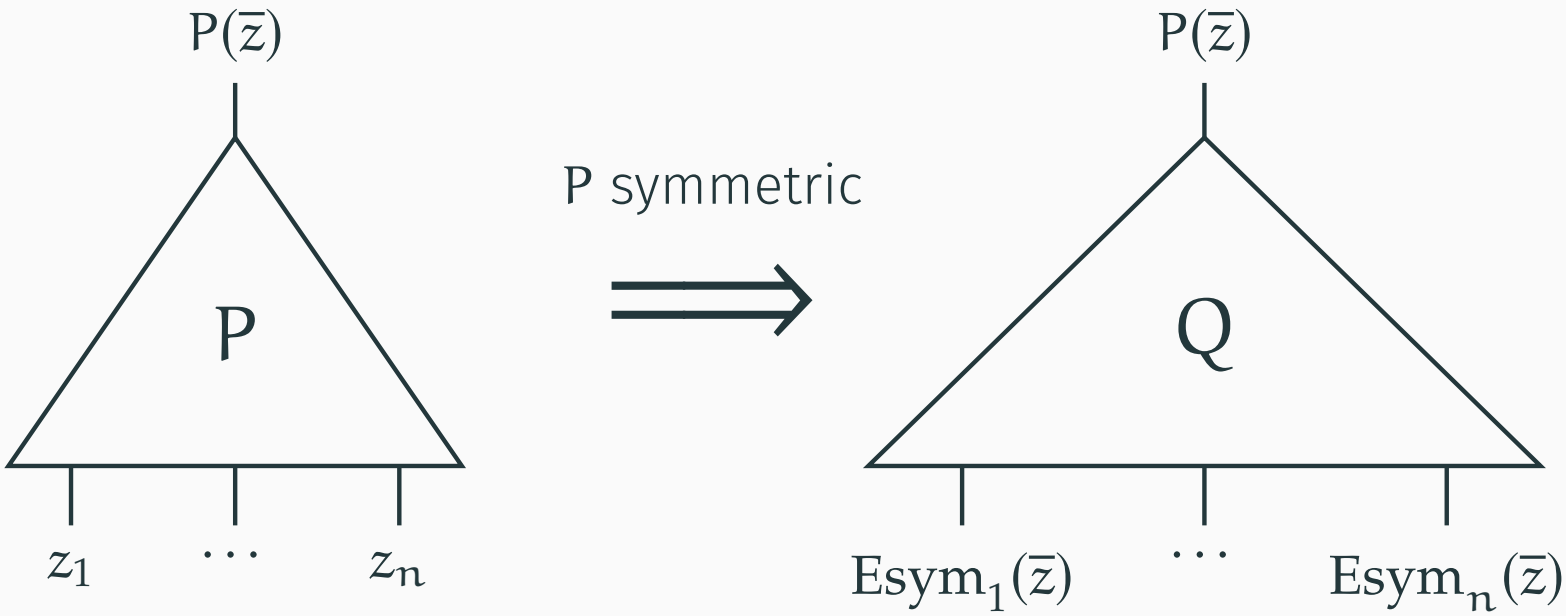
$f(z_1, \ldots, z_n)$ is a symmetric polynomial if for any permutation $\pi : [n] \to [n]$,

$$f(z_{\pi(1)}, \ldots, z_{\pi(n)}) = f(z_1, \ldots, z_n).$$

Example:

$$\mathrm{Esym}_d(\bar{z}) = \sum_{\substack{S \subseteq [n] \\ |S| = d}} \prod_{j \in S} z_j$$

Fundamental Theorem of Symmetric Polynomials



Q unique.

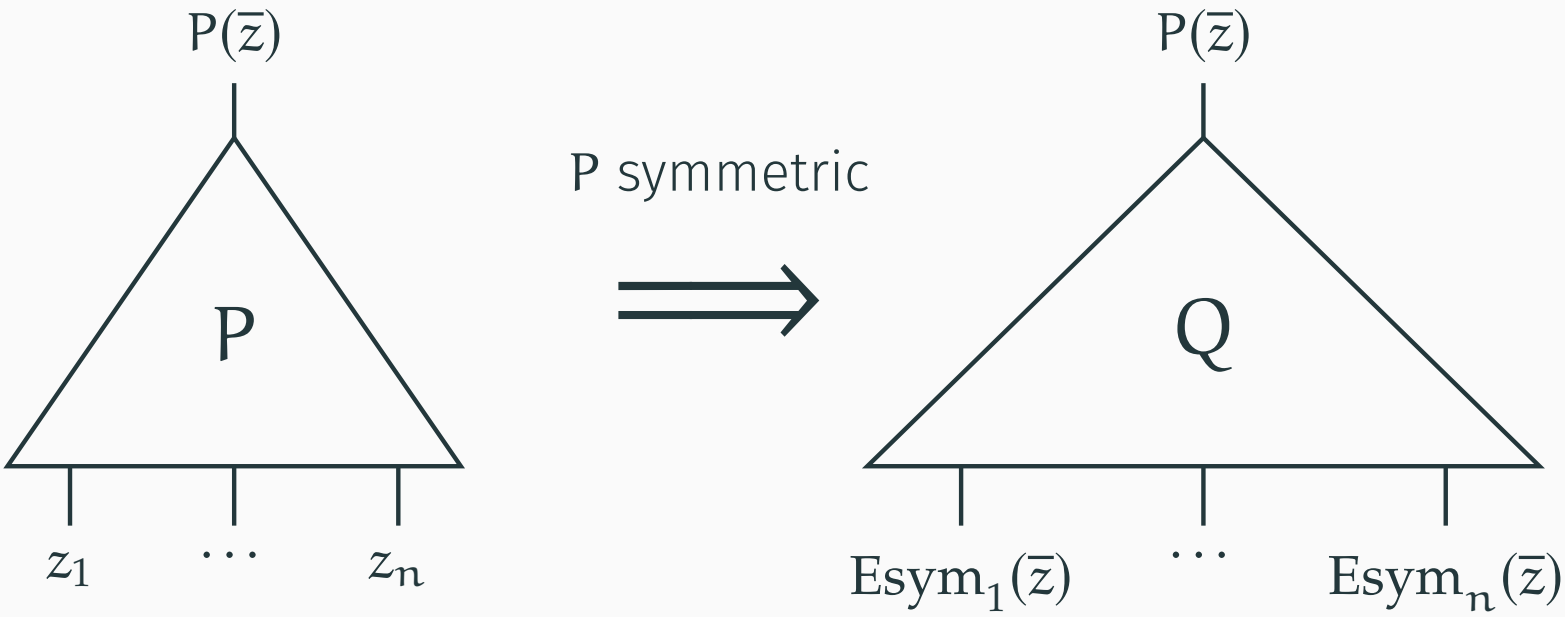
Vieta's theorem

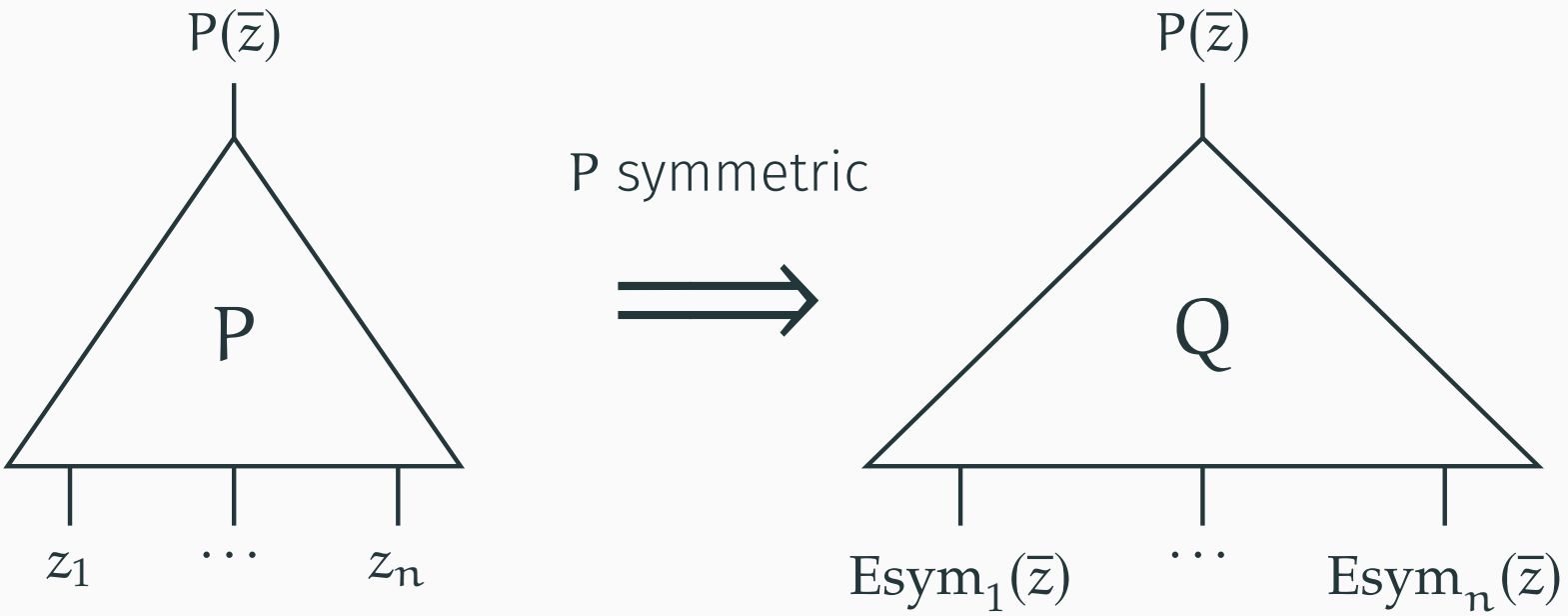
If

$$f(x) = \prod_{i \in [d]} (x - \alpha_i) = x^d + f_{d-1} x^{d-1} + \dots f_0$$

then for $i \in \{0, 1, \dots, d-1\}$,

$$f_i = (-1)^{d-i} \mathrm{Esym}_{d-i}(\alpha_1, \dots, \alpha_d)$$

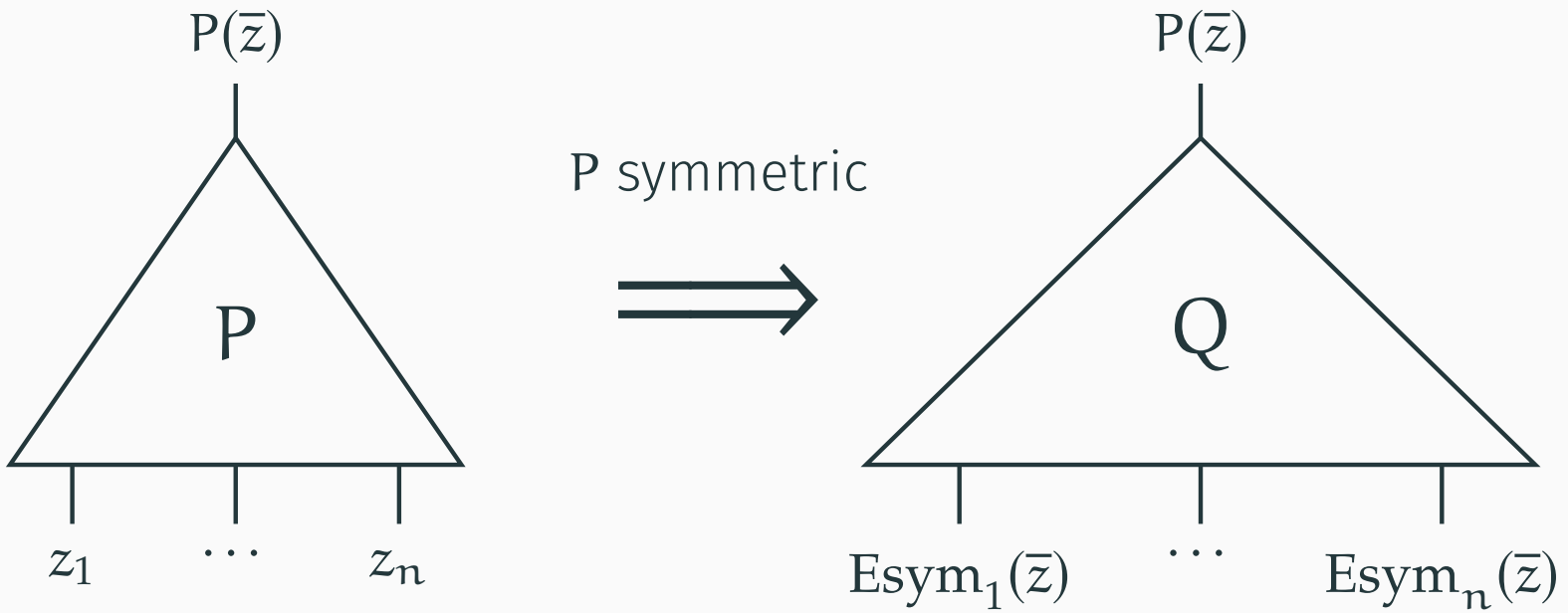

P(z̄)
P
z1
· · ·
zn
P symmetric
P(z̄)
Q
Esym1(z̄)
· · ·
Esymn(z̄)



size s, degree d

size $\mathrm{poly}(s, d, n)$

Our result — BKRRSS



size s, degree d, depth Δ

size $\mathrm{poly}(s, d, n)$, depth $\Delta + O(1)$

Over large enough fields. Also for multi-symmetric polynomials.

Resultant in constant-depth

Theorem (BKRRSS)

Over any large enough field, $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Proof.

Resultant in constant-depth

Theorem (BKRRSS)

Over any large enough field, $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Proof.

$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$: depth 2, size d^2.

Resultant in constant-depth

Theorem (BKRRSS)

Over any large enough field, $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Proof.

$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$: depth 2, size d^2.

Symmetric with respect to $(\alpha_1, \ldots, \alpha_d)$ and $(\beta_1, \ldots, \beta_d)$.

Resultant in constant-depth

Theorem (BKRRSS)

Over any large enough field, $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Proof.

$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$: depth 2, size d^2.

Symmetric with respect to $(\alpha_1, \ldots, \alpha_d)$ and $(\beta_1, \ldots, \beta_d)$. Thus,

$$\mathrm{Res}(f, g) = Q(\mathrm{Esym}_1(\overline{\alpha}), \ldots, \mathrm{Esym}_d(\overline{\alpha}), \mathrm{Esym}_1(\overline{\beta}), \ldots, \mathrm{Esym}_d(\overline{\beta}))$$

where Q has size $\mathrm{poly}(d)$, depth $O(1)$.

Resultant in constant-depth

Theorem (BKRRSS)

Over any large enough field, $\mathrm{Res}(f, g)$ *has size* $\mathrm{poly}(d)$*, depth* $O(1)$ *circuits.*

Proof.

$\mathrm{Res}(f, g) := \prod_{i \in [d], j \in [d]} (\alpha_i - \beta_j)$: depth 2, size d^2.

Symmetric with respect to $(\alpha_1, \ldots, \alpha_d)$ and $(\beta_1, \ldots, \beta_d)$. Thus,

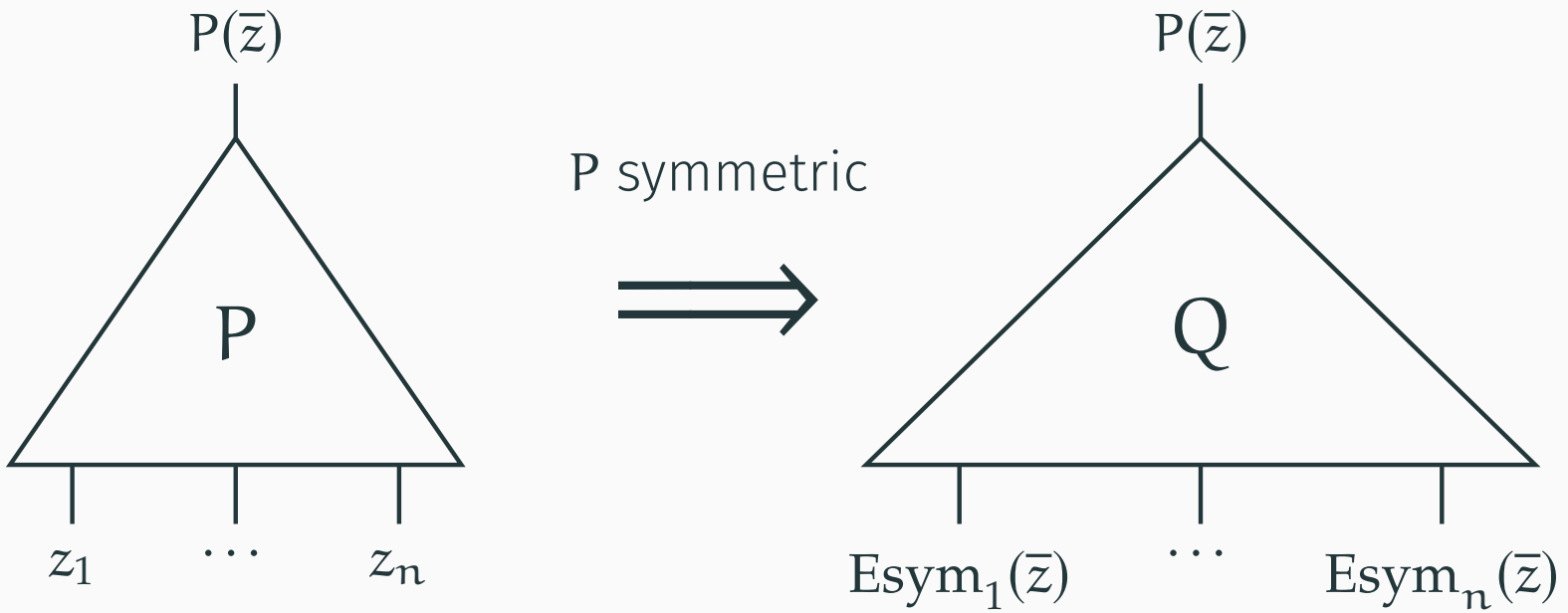
$$\mathrm{Res}(f, g) = Q(\mathrm{Esym}_1(\overline{\alpha}), \ldots, \mathrm{Esym}_d(\overline{\alpha}), \mathrm{Esym}_1(\overline{\beta}), \ldots, \mathrm{Esym}_d(\overline{\beta}))$$

where Q has size $\mathrm{poly}(d)$, depth $O(1)$.

Apply Vieta.

□

Our result — BKRRSS

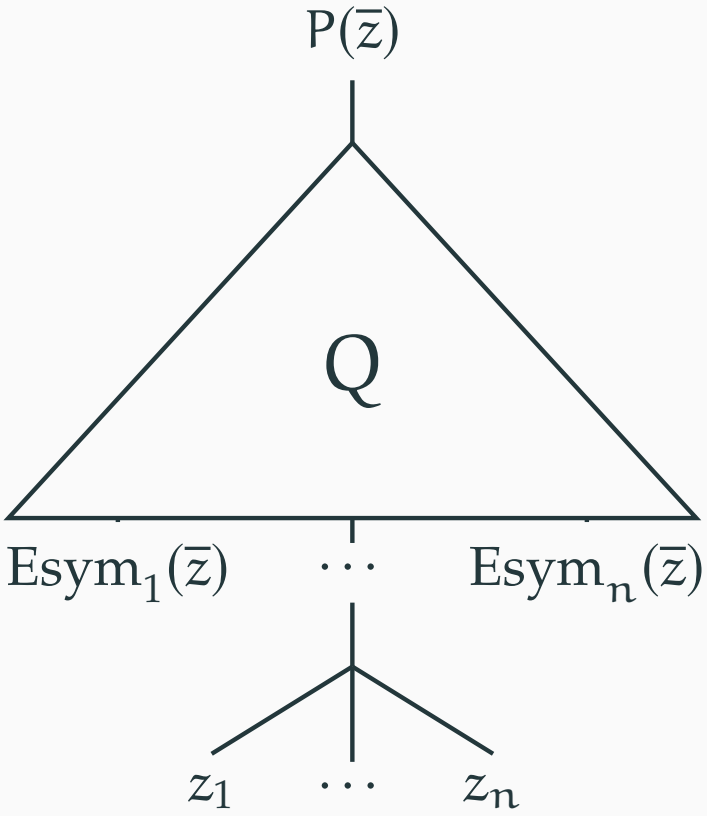


size s, degree d, depth Δ

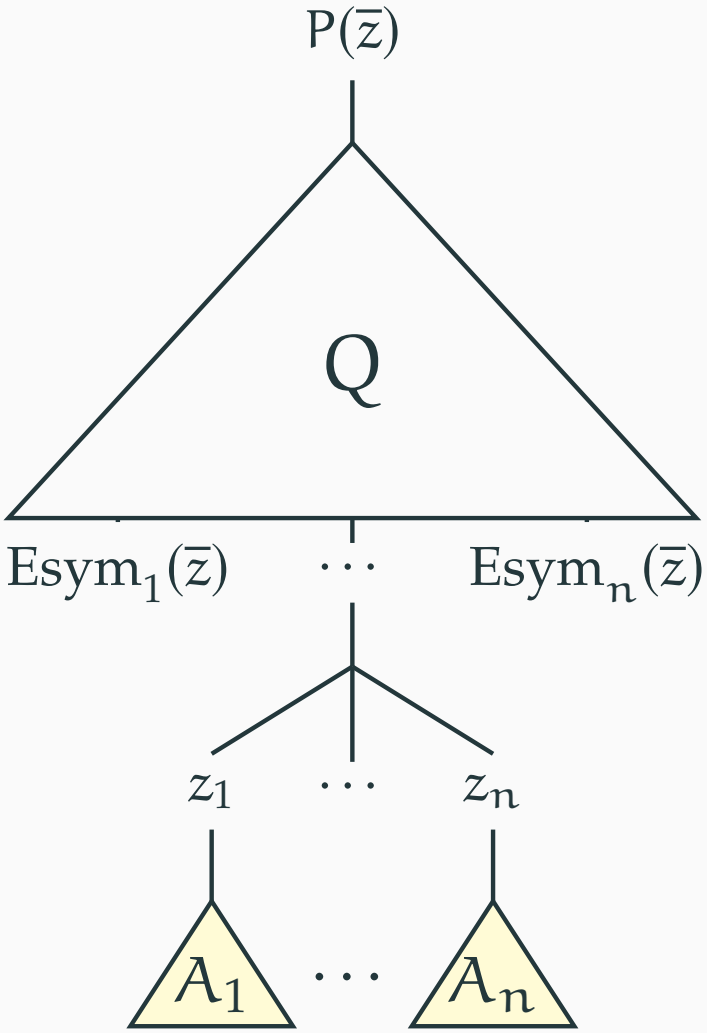
size $\mathrm{poly}(s, d, n)$, depth $\Delta + O(1)$

Over large enough fields. Also for multi-symmetric polynomials.

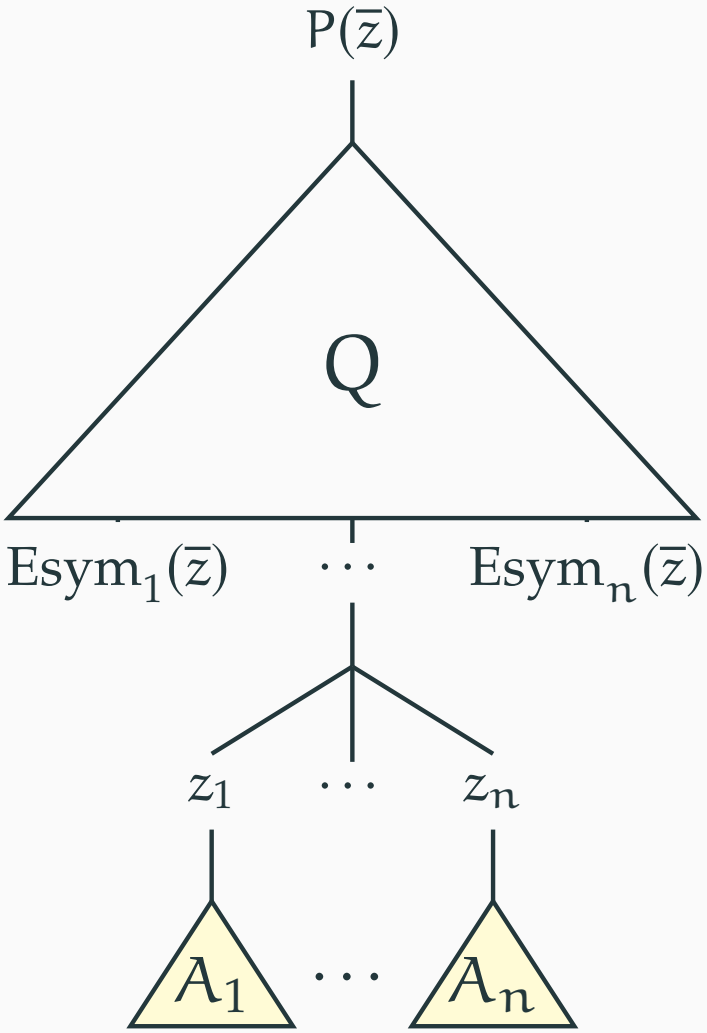
Proof idea

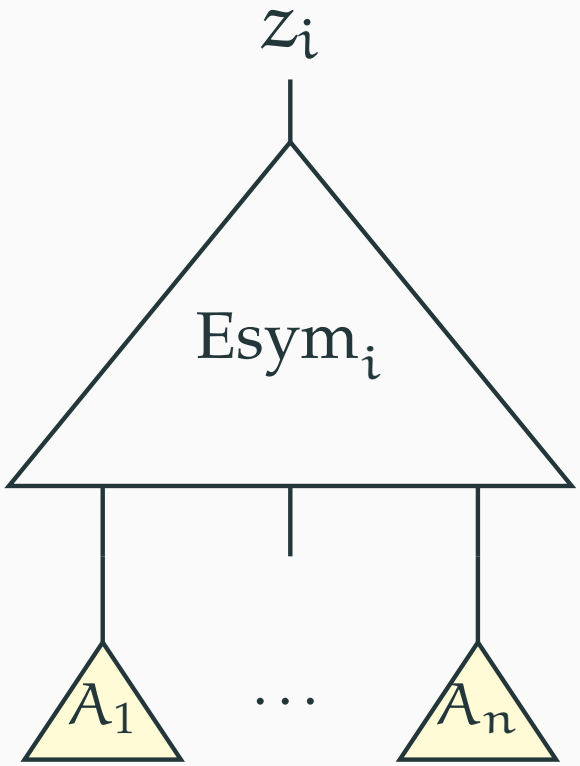

$P(\bar{z})$
Q
$\mathrm{Esym}_1(\bar{z})$
$\cdots$
$\mathrm{Esym}_n(\bar{z})$
z_1
$\cdots$
z_n

Proof idea


$P(\bar{z})$
Q
$\mathrm{Esym}_1(\bar{z})$
$\cdots$
$\mathrm{Esym}_n(\bar{z})$
z_1
$\cdots$
z_n
A_1
$\cdots$
A_n

Proof idea


$P(\bar{z})$
Q
$\mathrm{Esym}_1(\bar{z})$
$\cdots$
$\mathrm{Esym}_n(\bar{z})$
z_1
$\cdots$
z_n
A_1
$\cdots$
A_n


z_i
Esym_i
A_1
$\cdots$
A_n

Proof idea.

Proof idea.

Want to find efficient circuits $A_1(\bar{z}), \ldots, A_n(\bar{z})$ such that for every $i \in [n]$,

$$\mathrm{Esym}_i(A_1, \ldots, A_n) = z_i$$

Proof idea.

Want to find efficient circuits $A_1(\bar{z}), \ldots, A_n(\bar{z})$ such that for every $i \in [n]$,

$$\mathrm{Esym}_i(A_1, \ldots, A_n) = z_i$$

By Vieta, same as computing the roots of

$$R(\bar{z}, x) = x^n + z_1 x^{n-1} + z_2 x^{n-2} + \cdots + z_n$$

as a univariate in x.

Proof idea.

Want to find efficient circuits $A_1(\bar{z}), \ldots, A_n(\bar{z})$ such that for every $i \in [n]$,

$$\mathrm{Esym}_i(A_1, \ldots, A_n) = z_i$$

By Vieta, same as computing the roots of

$$R(\bar{z}, x) = x^n + z_1 x^{n-1} + z_2 x^{n-2} + \cdots + z_n$$

as a univariate in x.

Recent structural results for roots/factors of multivariate polynomials [BKRRSS25]: we can express these roots $A_i(\bar{z})$ using circuits of size $\mathrm{poly}(n)$ and depth $O(1)$.

□

Remarks

Remarks

Can extend to GCD using ideas similar to Andrews-Wigderson.

Remarks

Can extend to GCD using ideas similar to Andrews-Wigderson.

Works for other models like arithmetic formulas, arithmetic branching programs, etc.

Takeaway

Efficient symmetric expression using roots

$$\Longrightarrow$$

efficient expression using coefficients.

Open questions

Half-GCD algorithm: sequential near-linear time algorithm.
Constant-depth circuit of near-linear size for polynomial GCD?

Open questions

Half-GCD algorithm: sequential near-linear time algorithm. Constant-depth circuit of near-linear size for polynomial GCD?

We studied the algebra generated by elementary symmetric polynomials. What about the algebra generated by other sets of algebraically independent polynomials?

Open questions

Half-GCD algorithm: sequential near-linear time algorithm. Constant-depth circuit of near-linear size for polynomial GCD?

We studied the algebra generated by elementary symmetric polynomials. What about the algebra generated by other sets of algebraically independent polynomials?

Questions?