On the Universality of border width-2 ABPs over characteristic 2

Joint work with Pranjal Dutta, Balagopal Komarath, Harshil Mittal, and Saraswati Nanoti.

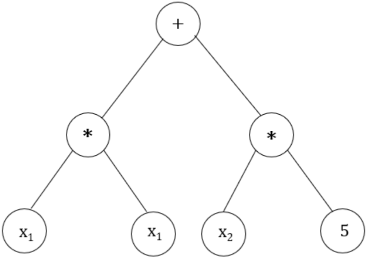
Dhara Thakkar
Indian Institute of Technology Gandhinagar, India.

31st March, 2023

Workshop on Algebraic Complexity Theory 2023

Table of Contents

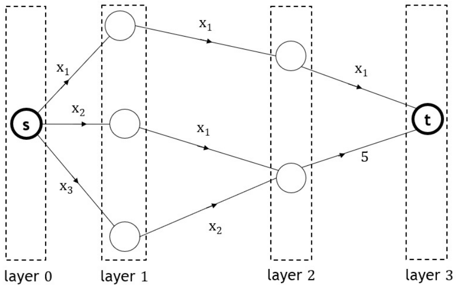
Arithmetic Formula



$x_1{}^2 + 5x_2$

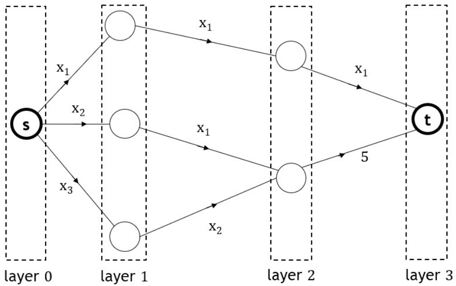
VF consists of families of polynomial with polynomially bounded formula size.

Algebraic Branching Programs (ABP)



VBP consists of families of polynomial that have ABPs of polynomially bounded size.

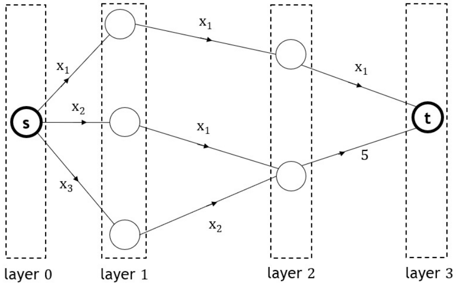
Algebraic Branching Programs (ABP)



VBP consists of families of polynomial that have ABPs of polynomially bounded size.

ABPs with restricted edge labels [BIZ, J. ACM, 2016]:

Algebraic Branching Programs (ABP)

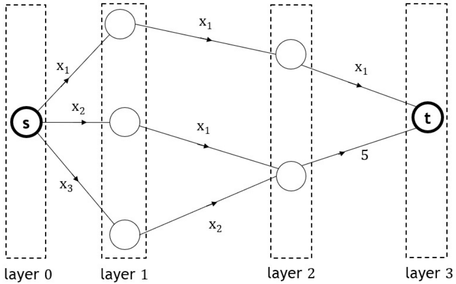


VBP consists of families of polynomial that have ABPs of polynomially bounded size.

ABPs with restricted edge labels [BIZ, J. ACM, 2016]:

- Weakest ABP: Edges are labeled by variables or constants.

Algebraic Branching Programs (ABP)

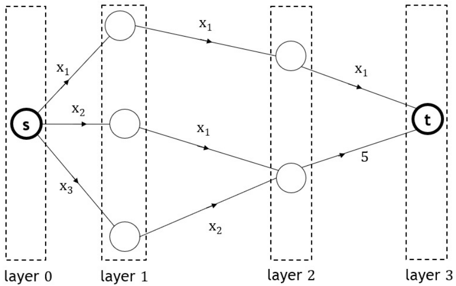


VBP consists of families of polynomial that have ABPs of polynomially bounded size.

ABPs with restricted edge labels [BIZ, J. ACM, 2016]:

- Weakest ABP: Edges are labeled by variables or constants.
- Weak ABP: Edges are labeled by simple affine linear forms $\alpha x_i + \beta, \alpha, \beta \in \mathcal{F}$.

Algebraic Branching Programs (ABP)



VBP consists of families of polynomial that have ABPs of polynomially bounded size.

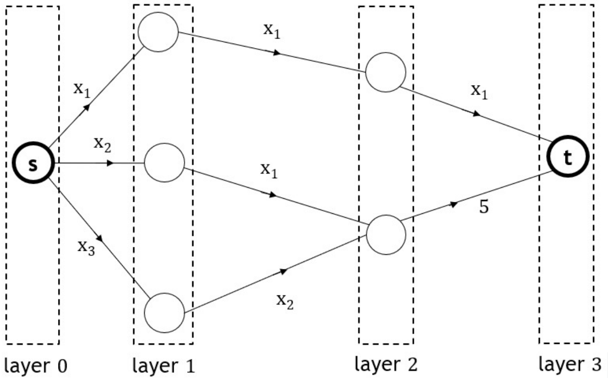
ABPs with restricted edge labels [BIZ, J. ACM, 2016]:

- Weakest ABP: Edges are labeled by variables or constants.
- Weak ABP: Edges are labeled by simple affine linear forms $\alpha x_i + \beta, \alpha, \beta \in \mathcal{F}$.
- We use Weakest ABP.

Width of an ABP

Width of an ABP

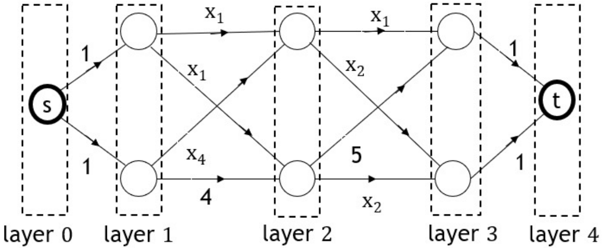
The *width* of an ABP is the maximum number of nodes in a layer.



Here, the width is 3.

ABPs and Matrix Multiplication

ABPs and Matrix Multiplication



$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_1 \\ x_4 & 4 \end{pmatrix} \begin{pmatrix} x_1 & x_2 \\ 5 & x_2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

This computes the polynomial $x_1^2 + 2x_1x_2 + x_2x_4 + x_1x_4 + 5x_1 + 4x_2 + 20$.

Background/Motivation

Background/Motivation

- ABPs are 'at least as powerful' as formulas.
  Precisely, $\mathsf{VF} \subseteq \mathsf{VBP}$.

Background/Motivation

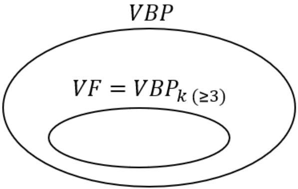
- ABPs are ‘at least as powerful’ as formulas.
  Precisely, VF $\subseteq$ VBP.
- What if we restrict the width of ABPs to some fixed $k \geq 3$?

Background/Motivation

- ABPs are ‘at least as powerful’ as formulas.
  Precisely, VF $\subseteq$ VBP.
- What if we restrict the width of ABPs to some fixed $k \geq 3$?
  VBP_k consists of families that have width-k ABPs of polynomially bounded size.

Background/Motivation

- ABPs are ‘at least as powerful’ as formulas.
  Precisely, VF $\subseteq$ VBP.
- What if we restrict the width of ABPs to some fixed $k \geq 3$?
  VBP_k consists of families that have width-k ABPs of polynomially bounded size.
  $\text{VBP}_k =$ VF [Ben-Or and Cleve, SIAM J. Comp., 1992].



Background/Motivation

- $\mathsf{VBP}_2 \stackrel{?}{=} \mathsf{VF}$.

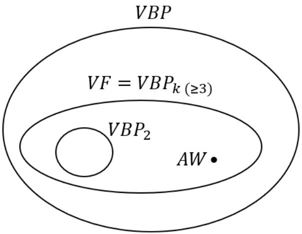
- $\mathrm{VBP}_2 \stackrel{?}{=} \mathrm{VF}$.
- The polynomial $\mathrm{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ cannot be computed by any width-2 ABP [Allender and Wang, CC, 2016].

Background/Motivation

- $\mathrm{VBP}_2 \stackrel{?}{=} \mathrm{VF}$.
- The polynomial $\mathrm{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ cannot be computed by any width-2 ABP [Allender and Wang, CC, 2016].
- $\mathrm{VBP}_2 \subsetneq \mathrm{VF}$.

Background/Motivation

- $\mathsf{VBP}_2 \stackrel{?}{=} \mathsf{VF}$.
- The polynomial $\mathsf{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ cannot be computed by any width-2 ABP [Allender and Wang, CC, 2016].
- $\mathsf{VBP}_2 \subsetneq \mathsf{VF}$.
  That is width-2 ABPs are 'strictly less powerful' than formulas.



Background/Motivation

- $\mathsf{VBP}_2 \stackrel{?}{=} \mathsf{VF}$.
- The polynomial $\mathsf{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ cannot be computed by any width-2 ABP [Allender and Wang, CC, 2016].
- $\mathsf{VBP}_2 \subsetneq \mathsf{VF}$.
  That is width-2 ABPs are 'strictly less powerful' than formulas.

[1] when $char(F) \neq 2$.

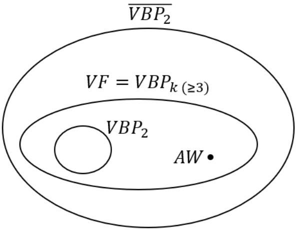
Background/Motivation

- $\mathsf{VBP}_2 \stackrel{?}{=} \mathsf{VF}$.
- The polynomial $\mathsf{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ cannot be computed by any width-2 ABP [Allender and Wang, CC, 2016].
- $\mathsf{VBP}_2 \subsetneq \mathsf{VF}$.
  That is width-2 ABPs are ‘strictly less powerful’ than formulas.
- What if we allow *‘approximation’*?

[1] when $char(F) \neq 2$.

Background/Motivation

- $\mathsf{VBP}_2 \stackrel{?}{=} \mathsf{VF}$.
- The polynomial $\mathsf{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ cannot be computed by any width-2 ABP [Allender and Wang, CC, 2016].
- $\mathsf{VBP}_2 \subsetneq \mathsf{VF}$.
  That is width-2 ABPs are 'strictly less powerful' than formulas.
- What if we allow *'approximation'*?
  Then, they become 'at least as powerful' as formulas[1].



[1] when $char(F) \neq 2$.

Algebraic Approximation

Algebraic Approximation

Definition: A polynomial $g \in \mathbb{F}(\epsilon)[X]$ *approximates* $f \in \mathbb{F}[X]$ with *error degree* e if $g = f + \epsilon h_1 + \epsilon^2 h_2 + \cdots + \epsilon^e h_e$ where each $h_i \in \mathbb{F}[X]$.

Algebraic Approximation

Definition: A polynomial $g \in \mathbb{F}(\epsilon)[X]$ *approximates* $f \in \mathbb{F}[X]$ with *error degree* e if $g = f + \epsilon h_1 + \epsilon^2 h_2 + \cdots + \epsilon^e h_e$ where each $h_i \in \mathbb{F}[X]$.

Definition: The *approximation closure* $\overline{C}(\mathbb{F})$ consists of families (f_n) for which there exists a family $(g_n) \in C(\mathbb{F}(\epsilon))$ such that g_n approximates f_n for all n.

Algebraic Approximation

Definition: A polynomial $g \in \mathbb{F}(\epsilon)[X]$ *approximates* $f \in \mathbb{F}[X]$ with *error degree* e if $g = f + \epsilon h_1 + \epsilon^2 h_2 + \cdots + \epsilon^e h_e$ where each $h_i \in \mathbb{F}[X]$.

Definition: The *approximation closure* $\overline{C}(\mathbb{F})$ consists of families (f_n) for which there exists a family $(g_n) \in C(\mathbb{F}(\epsilon))$ such that g_n approximates f_n for all n.

In addition, if we restrict the error degree to be polynomialy bounded in n then the class obtained is called *poly-approximation closure*, denoted as $\overline{C}^{\text{poly}}(\mathbb{F})$.

Algebraic Approximation

Definition: A polynomial $g \in \mathbb{F}(\epsilon)[X]$ *approximates* $f \in \mathbb{F}[X]$ with *error degree* e if $g = f + \epsilon h_1 + \epsilon^2 h_2 + \cdots + \epsilon^e h_e$ where each $h_i \in \mathbb{F}[X]$.

Definition: The *approximation closure* $\overline{C}(\mathbb{F})$ consists of families (f_n) for which there exists a family $(g_n) \in C(\mathbb{F}(\epsilon))$ such that g_n approximates f_n for all n.

In addition, if we restrict the error degree to be polynomialy bounded in n then the class obtained is called *poly-approximation closure*, denoted as $\overline{C}^{\mathsf{poly}}(\mathbb{F})$.

- $\mathsf{AW} \notin \mathsf{VBP}_2 \subsetneq \mathsf{VF}$.

Algebraic Approximation

Definition: A polynomial $g \in \mathbb{F}(\epsilon)[X]$ *approximates* $f \in \mathbb{F}[X]$ with *error degree* e if $g = f + \epsilon h_1 + \epsilon^2 h_2 + \cdots + \epsilon^e h_e$ where each $h_i \in \mathbb{F}[X]$.

Definition: The *approximation closure* $\overline{C}(\mathbb{F})$ consists of families (f_n) for which there exists a family $(g_n) \in C(\mathbb{F}(\epsilon))$ such that g_n approximates f_n for all n.

In addition, if we restrict the error degree to be polynomialy bounded in n then the class obtained is called *poly-approximation closure*, denoted as $\overline{C}^{\mathsf{poly}}(\mathbb{F})$.

- $\mathrm{AW} \notin \mathrm{VBP}_2 \subsetneq \mathrm{VF}$.
- Is $\mathrm{AW} \in \overline{\mathrm{VBP}_2}$?

Approximation Helps

The polynomial $\mathrm{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ can be approximated by width-2 ABP when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J.ACM 2018].

Approximation Helps

The polynomial AW $= \sum_{i=1}^{8} x_{2i-1}x_{2i}$ can be approximated by width-2 ABP when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J.ACM 2018].

Define $F(x, y) :=$

$$\begin{pmatrix} \frac{1}{\epsilon} - \frac{\epsilon x}{2} & -\frac{x}{2\epsilon} \\ \epsilon^3 & \epsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2}(x-2y)\epsilon^2 + 1 & \frac{1}{2}(x-2y) \\ \epsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x\epsilon^2}{2} + 1 & -\frac{x}{2} \\ -\epsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x+2y}{2\epsilon} & \epsilon \\ \epsilon^{-1} & 0 \end{pmatrix}$$

Approximation Helps

The polynomial AW $= \sum_{i=1}^{8} x_{2i-1}x_{2i}$ can be approximated by width-2 ABP when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J.ACM 2018].

Define $F(x, y) :=$

$$\begin{pmatrix} \frac{1}{\epsilon} - \frac{\epsilon x}{2} & -\frac{x}{2\epsilon} \\ \epsilon^3 & \epsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2}(x-2y)\epsilon^2 + 1 & \frac{1}{2}(x-2y) \\ \epsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x\epsilon^2}{2} + 1 & -\frac{x}{2} \\ -\epsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x+2y}{2\epsilon} & \epsilon \\ \epsilon^{-1} & 0 \end{pmatrix}$$

$$F(x, y) = \begin{pmatrix} xy + \mathcal{O}(\epsilon) & 1 + \mathcal{O}(\epsilon) \\ 1 + \mathcal{O}(\epsilon) & 0 + \mathcal{O}(\epsilon) \end{pmatrix} = \begin{pmatrix} xy & 1 \\ 1 & 0 \end{pmatrix} + \mathcal{O}(\epsilon).$$

Approximation Helps

The polynomial $\mathrm{AW} = \sum_{i=1}^{8} x_{2i-1}x_{2i}$ can be approximated by width-2 ABP when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J.ACM 2018].

Define $F(x,y) :=$

$$\begin{pmatrix} \frac{1}{\epsilon} - \frac{\epsilon x}{2} & -\frac{x}{2\epsilon} \\ \epsilon^3 & \epsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2}(x-2y)\epsilon^2 + 1 & \frac{1}{2}(x-2y) \\ \epsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x\epsilon^2}{2} + 1 & -\frac{x}{2} \\ -\epsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x+2y}{2\epsilon} & \epsilon \\ \epsilon^{-1} & 0 \end{pmatrix}$$

$$F(x,y) = \begin{pmatrix} xy + \mathcal{O}(\epsilon) & 1 + \mathcal{O}(\epsilon) \\ 1 + \mathcal{O}(\epsilon) & 0 + \mathcal{O}(\epsilon) \end{pmatrix} = \begin{pmatrix} xy & 1 \\ 1 & 0 \end{pmatrix} + \mathcal{O}(\epsilon).$$

The following sequence approximately computes AW

$$\begin{pmatrix} 1 & 0 \end{pmatrix} F(x_1, x_2) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} F(x_{15}, x_{16}) \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Approximately computing AW when $\mathrm{char}(\mathbb{F}) = 2$

Question: Can AW be computed approximately even when $char(\mathbb{F}) = 2$?

Yes!

Approximately computing AW when $\mathrm{char}(\mathbb{F}) = 2$

Question: Can AW be computed approximately even when $char(\mathbb{F}) = 2$?

Yes!

Define $F(x, y) := \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & y \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}.$

Approximately computing AW when $\mathrm{char}(\mathbb{F}) = 2$

Question: Can AW be computed approximately even when $char(\mathbb{F}) = 2$?
Yes!

Define $F(x, y) := \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & y \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}$.

$F(x, y)$ computes $\begin{pmatrix} xy & 1 \\ 1 + \epsilon y & 0 \end{pmatrix}$.

Question: Can AW be computed approximately even when $char(\mathbb{F}) = 2$?

Yes!

Define $F(x, y) := \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & y \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}$.

$F(x, y)$ computes $\begin{pmatrix} xy & 1 \\ 1 + \epsilon y & 0 \end{pmatrix}$.

The following sequence approximately computes AW

$$\begin{pmatrix} 1 & 0 \end{pmatrix} F(x_1, x_2) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} F(x_{15}, x_{16}) \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Approximately computing AW when $\mathrm{char}(\mathbb{F}) = 2$

Question: Can AW be computed approximately even when $char(\mathbb{F}) = 2$?
Yes!

Define $F(x, y) := \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & y \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}$.

$F(x, y)$ computes $\begin{pmatrix} xy & 1 \\ 1 + \epsilon y & 0 \end{pmatrix}$.

The following sequence approximately computes AW

$$\begin{pmatrix} 1 & 0 \end{pmatrix} F(x_1, x_2) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} F(x_{15}, x_{16}) \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Note: This is true for arbitrary field.

- $\mathsf{AW} \in \overline{\mathsf{VBP}_2}$

- $\mathsf{AW} \in \overline{\mathsf{VBP}_2}$
- $\mathsf{VF} \overset{?}{\subseteq} \overline{\mathsf{VBP}_2}$.

- $\mathrm{AW} \in \overline{\mathrm{VBP}_2}$
- $\mathrm{VF} \overset{?}{\subseteq} \overline{\mathrm{VBP}_2}$.
- If we allow for approximation, it becomes ‘at least as powerful’ as formulas.

- $\mathrm{AW} \in \overline{\mathrm{VBP}_2}$
- $\mathrm{VF} \overset{?}{\subseteq} \overline{\mathrm{VBP}_2}$.
- If we allow for approximation, it becomes 'at least as powerful' as formulas. Precisely, $\mathrm{VF} \subseteq \overline{\mathrm{VBP}_2}$ when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J. ACM, 2016].

- $\mathsf{AW} \in \overline{\mathsf{VBP}_2}$
- $\mathsf{VF} \overset{?}{\subseteq} \overline{\mathsf{VBP}_2}$.
- If we allow for approximation, it becomes 'at least as powerful' as formulas. Precisely, $\mathsf{VF} \subseteq \overline{\mathsf{VBP}_2}$ when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J. ACM, 2016].
- $\overline{\mathsf{VF}} = \overline{\mathsf{VBP}_2}$ when $\mathrm{char}(\mathbb{F}) \neq 2$ [BIZ, J. ACM, 2016].

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

- Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

- Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

- $Q(f) := \begin{pmatrix} f & 1 \\ 1 & 0 \end{pmatrix}$.

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

- Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

- $Q(f) := \begin{pmatrix} f & 1 \\ 1 & 0 \end{pmatrix}.$
- $Q(f+g) = Q(f)Q(0)Q(g)$

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

- Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

- $Q(f) := \begin{pmatrix} f & 1 \\ 1 & 0 \end{pmatrix}$.
- $Q(f+g) = Q(f)Q(0)Q(g)$
- $Q(f \cdot g)$?

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

* Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

* $Q(f) := \begin{pmatrix} f & 1 \\ 1 & 0 \end{pmatrix}$.
* $Q(f+g) = Q(f)Q(0)Q(g)$
* $Q(f \cdot g)$?

$$f \cdot g = (f/2 + g)^2 + (-(f/2)^2) + (-g^2)$$

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

- Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

- $Q(f) := \begin{pmatrix} f & 1 \\ 1 & 0 \end{pmatrix}$.
- $Q(f+g) = Q(f)Q(0)Q(g)$
- $Q(f \cdot g)$?

$$f \cdot g = (f/2 + g)^2 + (-(f/2)^2) + (-g^2)$$

- $Q(f^2 + \mathcal{O}(\epsilon)) = Q(-\epsilon^{-1}) \cdot Q(\epsilon) \cdot Q(-\epsilon^{-1}) \cdot (Q(f + \mathcal{O}(\epsilon^3))) \cdot Q(1) \cdot Q(-1) \cdot Q(1) \cdot Q(-\epsilon^2) \cdot (Q(f + \mathcal{O}(\epsilon^3))) \cdot Q(-\epsilon^{-1}) \cdot Q(\epsilon - 1) \cdot Q(1) \cdot Q(\epsilon^{-1} - 1)$

Theorem (BIZ, J. ACM, 2016)

$\mathsf{VF} \subset \overline{\mathsf{VBP}_2}^{poly}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof Idea:

- Use induction on the depth to show

$$\text{depth } d\text{- formula} \implies \text{sequence of } 2^{\mathcal{O}(d)} \text{ matrices.}$$

- $Q(f) := \begin{pmatrix} f & 1 \\ 1 & 0 \end{pmatrix}.$
- $Q(f+g) = Q(f)Q(0)Q(g)$
- $Q(f \cdot g)$?

$$f \cdot g = (f/2 + g)^2 + (-(f/2)^2) + (-g^2)$$

- $Q(f^2 + \mathcal{O}(\epsilon)) = Q(-\epsilon^{-1}) \cdot Q(\epsilon) \cdot Q(-\epsilon^{-1}) \cdot (Q(f + \mathcal{O}(\epsilon^3))) \cdot Q(1) \cdot Q(-1) \cdot Q(1) \cdot Q(-\epsilon^2) \cdot (Q(f + \mathcal{O}(\epsilon^3))) \cdot Q(-\epsilon^{-1}) \cdot Q(\epsilon - 1) \cdot Q(1) \cdot Q(\epsilon^{-1} - 1)$

Thus, $\overline{\mathsf{VF}} = \overline{\mathsf{VBP}_2}$ when $\mathrm{char}(\mathbb{F}) \neq 2$

- Is $\mathsf{VF} \subset \overline{\mathsf{VBP}_2}$ even when $\mathrm{char}(\mathbb{F}) = 2$?

- Is $\mathrm{VF} \subset \overline{\mathrm{VBP}_2}$ even when $\mathrm{char}(\mathbb{F}) = 2$?
  We still do not know!

- Is $\mathsf{VF} \subset \overline{\mathsf{VBP}_2}$ even when $\mathrm{char}(\mathbb{F}) = 2$?
  We still do not know!
- Is the Model universal? .... (i.e. can every polynomial be approximately computed using width-2 ABP (or seq. of 2×2 matrices?)

- Is $\mathsf{VF} \subset \overline{\mathsf{VBP}_2}$ even when $\mathrm{char}(\mathbb{F}) = 2$?
  We still do not know!
- Is the Model universal? .... (i.e. can every polynomial be approximately computed using width-2 ABP (or seq. of 2×2 matrices?)
  Yes!

Lemma 1

Let f be a polynomial. Suppose that there is a sequence, say σ, of N matrices that approximately computes $Q(f)$. Then, for any indeterminate x, there is a sequence of $2N+4$ matrices that approximately computes $Q(fx)$.

Lemma 1
Let f be a polynomial. Suppose that there is a sequence, say σ, of N matrices that approximately computes $Q(f)$. Then, for any indeterminate x, there is a sequence of $2N+4$ matrices that approximately computes $Q(fx)$.

Proof Sketch:

$$Q(fx) + \mathcal{O}(\epsilon) = \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & x \\ -1 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}.$$

Showing Universality

Lemma 1

Let f be a polynomial. Suppose that there is a sequence, say σ, of N matrices that approximately computes $Q(f)$. Then, for any indeterminate x, there is a sequence of $2N+4$ matrices that approximately computes $Q(fx)$.

Proof Sketch:

$$Q(fx) + \mathcal{O}(\epsilon) = \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & x \\ -1 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}.$$

- Although not as powerful as multiplying two arbitrary polynomials, this proves universality.

Lemma 1

Let f be a polynomial. Suppose that there is a sequence, say σ, of N matrices that approximately computes $Q(f)$. Then, for any indeterminate x, there is a sequence of $2N+4$ matrices that approximately computes $Q(fx)$.

Proof Sketch:

$$Q(fx) + \mathcal{O}(\epsilon) = \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & x \\ -1 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}.$$

- Although not as powerful as multiplying two arbitrary polynomials, this proves universality.
- However, this is *inefficient*.

  For example, consider x^n, we require $\mathcal{O}(2^n)$ matrices to compute $Q(x^n)$.

Showing Universality

Lemma 1
Let f be a polynomial. Suppose that there is a sequence, say σ, of N matrices that approximately computes $Q(f)$. Then, for any indeterminate x, there is a sequence of $2N+4$ matrices that approximately computes $Q(fx)$.

Proof Sketch:

$$Q(fx) + \mathcal{O}(\epsilon) = \begin{pmatrix} \frac{1}{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} \epsilon & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\epsilon} & x \\ -1 & 1 \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 1 & -\epsilon \end{pmatrix}.$$

- Although not as powerful as multiplying two arbitrary polynomials, this proves universality.
- However, this is *inefficient*.
  For example, consider x^n, we require $\mathcal{O}(2^n)$ matrices to compute $Q(x^n)$.
- We improve this to $\mathcal{O}(n)$.

Improving efficiency

Lemma 2
Let f and g be polynomials. Suppose that there are sequences, say σ and π, of N and M matrices, that approximately compute $Q(f)$ and $Q(g)$ respectively. Then, there is a sequence of $N + 2M + 4$ matrices that approximately computes $Q(fg^2)$.

Improving efficiency

Lemma 2
Let f and g be polynomials. Suppose that there are sequences, say σ and π, of N and M matrices, that approximately compute $Q(f)$ and $Q(g)$ respectively. Then, there is a sequence of $N + 2M + 4$ matrices that approximately computes $Q(fg^2)$.

Proof Sketch:

$$Q(fg^2) + \mathcal{O}(\epsilon) = \begin{pmatrix} -1 & 0 \\ 0 & \epsilon \end{pmatrix} \pi \,|_{\epsilon \to \epsilon^{\mathbf{2}}} \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\epsilon} \end{pmatrix} \sigma \,|_{\epsilon \to \epsilon^{\mathbf{3}}} \begin{pmatrix} -1 & 0 \\ 0 & \frac{1}{\epsilon} \end{pmatrix} \pi \,|_{\epsilon \to \epsilon^{\mathbf{2}}} \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}.$$

Lemma 2
Let f and g be polynomials. Suppose that there are sequences, say σ and π, of N and M matrices, that approximately compute $Q(f)$ and $Q(g)$ respectively. Then, there is a sequence of $N + 2M + 4$ matrices that approximately computes $Q(fg^2)$.

Proof Sketch:

$$Q(fg^2) + \mathcal{O}(\epsilon) = \begin{pmatrix} -1 & 0 \\ 0 & \epsilon \end{pmatrix} \pi \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\epsilon} \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^3} \begin{pmatrix} -1 & 0 \\ 0 & \frac{1}{\epsilon} \end{pmatrix} \pi \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}.$$

Theorem

Let p be a polynomial with ℓ monomials, each containing at most t indeterminates. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(\ell \cdot (2^t + \mathit{degree}(p)))$ matrices.

Improving efficiency

Lemma 2

Let f and g be polynomials. Suppose that there are sequences, say σ and π, of N and M matrices, that approximately compute $Q(f)$ and $Q(g)$ respectively. Then, there is a sequence of $N + 2M + 4$ matrices that approximately computes $Q(fg^2)$.

Proof Sketch:

$$Q(fg^2) + \mathcal{O}(\epsilon) = \begin{pmatrix} -1 & 0 \\ 0 & \epsilon \end{pmatrix} \pi \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\epsilon} \end{pmatrix} \sigma \mid_{\epsilon \to \epsilon^3} \begin{pmatrix} -1 & 0 \\ 0 & \frac{1}{\epsilon} \end{pmatrix} \pi \mid_{\epsilon \to \epsilon^2} \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}.$$

Theorem

Let p be a polynomial with ℓ monomials, each containing at most t indeterminates. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(\ell \cdot (2^t + degree(p)))$ matrices.

Observe that x^n needs only $O(n)$ matrices.

Improving efficiency cont.

Theorem:
Sequence of $\mathcal{O}(\ell \cdot (2^t + degree(p)))$ matrices that approximately computes $Q(p)$ where,
$\ell =$ no. of monomials in p
$t =$ max no. of indeterminates per monomial

Improving efficiency cont.

Theorem:
Sequence of $\mathcal{O}(\ell \cdot (2^t + \mathit{degree}(p)))$ matrices that approximately computes $Q(p)$ where,
$\ell =$ no. of monomials in p
$t =$ max no. of indeterminates per monomial

Proof Idea:

- Goal: Compute each monomial using at most $\mathcal{O}(2^t + \mathit{degree}(p))$ matrices.

Improving efficiency cont.

> **Theorem:**
> Sequence of $\mathcal{O}(\ell \cdot (2^t + degree(p)))$ matrices that approximately computes $Q(p)$ where,
> ℓ = no. of monomials in p
> t = max no. of indeterminates per monomial

Proof Idea:

- Goal: Compute each monomial using at most $\mathcal{O}(2^t + degree(p))$ matrices.
- Let $\underbrace{X_1^{r_1} \cdots X_k^{r_k}}_{\text{odd } r_i' \text{s}} \underbrace{X_{k+1}^{r_{k+1}} \cdots X_m^{r_m}}_{\text{even } r_i' \text{s}}$

Improving efficiency cont.

Theorem:
Sequence of $\mathcal{O}(\ell \cdot (2^t + degree(p)))$ matrices that approximately computes $Q(p)$ where,
$\ell =$ no. of monomials in p
$t =$ max no. of indeterminates per monomial

Proof Idea:

- Goal: Compute each monomial using at most $\mathcal{O}(2^t + degree(p))$ matrices.
- Let $\underbrace{X_1^{r_1} \cdots X_k^{r_k}}_{\text{odd } r_i's} \underbrace{X_{k+1}^{r_{k+1}} \cdots X_m^{r_m}}_{\text{even } r_i's} = (X_1 \cdots X_k) \cdot (X_1^{r_1 - 1} \cdots X_k^{r_k - 1} X_{k+1}^{r_{k+1}} \cdots X_m^{r_m})$

Improving efficiency cont.

Theorem:
Sequence of $\mathcal{O}(\ell \cdot (2^t + degree(p)))$ matrices that approximately computes $Q(p)$ where,
$\ell =$ no. of monomials in p
$t =$ max no. of indeterminates per monomial

Proof Idea:

- Goal: Compute each monomial using at most $\mathcal{O}(2^t + degree(p))$ matrices.
- Let $\underbrace{X_1^{r_1} \cdots X_k^{r_k}}_{\text{odd } r_i'\text{s}} \underbrace{X_{k+1}^{r_{k+1}} \cdots X_m^{r_m}}_{\text{even } r_i'\text{s}} = (X_1 \cdots X_k) \cdot (X_1^{r_1 - 1} \cdots X_k^{r_k - 1} X_{k+1}^{r_{k+1}} \cdots X_m^{r_m})$

Corollary: Let p be a univariate polynomial of degree d. Then $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d^2)$ matrices.

Improving efficiency cont.

> **Theorem:**
> Sequence of $\mathcal{O}(\ell \cdot (2^t + degree(p)))$ matrices that approximately computes $Q(p)$ where,
> $\ell =$ no. of monomials in p
> $t =$ max no. of indeterminates per monomial

Proof Idea:

- Goal: Compute each monomial using at most $\mathcal{O}(2^t + degree(p))$ matrices.
- Let $\underbrace{X_1^{r_1} \cdots X_k^{r_k}}_{\text{odd } r_i'\text{s}} \underbrace{X_{k+1}^{r_{k+1}} \cdots X_m^{r_m}}_{\text{even } r_i'\text{s}} = (X_1 \cdots X_k) \cdot (X_1^{r_1 - 1} \cdots X_k^{r_k - 1} X_{k+1}^{r_{k+1}} \cdots X_m^{r_m})$

Corollary: Let p be a univariate polynomial of degree d. Then $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d^2)$ matrices.

Note: However, we can do better!

Improving efficiency Further for univariate polynomial

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1} x^{d-1} + \ldots\ldots + a_2 x^2 + a_1 x + a_0$. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d)$ matrices.

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1} x^{d-1} + \ldots\ldots + a_2 x^2 + a_1 x + a_0$. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d)$ matrices.

Idea: Use Horner's rule

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1}x^{d-1} + \ldots\ldots + a_2x^2 + a_1x + a_0$. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d)$ matrices.

Idea: Use Horner's rule

Illustration: Consider a polynomial

$$a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = (a_4x^2 + a_3x + a_2)x^2 + a_1x + a_0.$$

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1} x^{d-1} + \ldots\ldots + a_2 x^2 + a_1 x + a_0$*. Then,* $Q(p)$ *can be approximately computed using a sequence of at most* $\mathcal{O}(d)$ *matrices.*

Idea: Use Horner's rule

Illustration: Consider a polynomial

$$a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = (a_4 x^2 + a_3 x + a_2) x^2 + a_1 x + a_0.$$

$$Q(a_4) \xrightarrow{\text{Lemma 2}} Q(a_4 x^2)$$

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1}x^{d-1} + \ldots\ldots + a_2x^2 + a_1x + a_0$. *Then,* $Q(p)$ *can be approximately computed using a sequence of at most* $\mathcal{O}(d)$ *matrices.*

Idea: Use Horner's rule

Illustration: Consider a polynomial

$$a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = (a_4x^2 + a_3x + a_2)x^2 + a_1x + a_0.$$

$$Q(a_4) \xrightarrow{\text{Lemma 2}} Q(a_4x^2)$$

$$\Big\downarrow \text{append } Q(0)\begin{pmatrix} a_3 & a_2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x & \frac{1}{a_3} \\ 1 & 0 \end{pmatrix}$$

$$Q(a_4x^2 + a_3x + a_2)$$

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1}x^{d-1} + \ldots\ldots + a_2x^2 + a_1x + a_0$. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d)$ matrices.

Idea: Use Horner's rule

Illustration: Consider a polynomial

$$a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = (a_4x^2 + a_3x + a_2)x^2 + a_1x + a_0.$$

$$Q(a_4) \xrightarrow{\text{Lemma 2}} Q(a_4x^2)$$

$$\Big\downarrow \text{append } Q(0) \begin{pmatrix} a_3 & a_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x & \frac{1}{a_3} \\ 1 & 0 \end{pmatrix}$$

$$Q(a_4x^2 + a_3x + a_2) \xrightarrow{\text{Lemma 2}} Q((a_4x^2 + a_3x + a_2)x^2)$$

Improving efficiency Further for univariate polynomial

Theorem

Consider a degree-d univariate polynomial in x, say $p = a_d x^d + a_{d-1}x^{d-1} + \ldots\ldots + a_2x^2 + a_1x + a_0$. Then, $Q(p)$ can be approximately computed using a sequence of at most $\mathcal{O}(d)$ matrices.

Idea: Use Horner's rule

Illustration: Consider a polynomial

$$a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = (a_4x^2 + a_3x + a_2)x^2 + a_1x + a_0.$$

$$Q(a_4) \xrightarrow{\text{Lemma 2}} Q(a_4x^2)$$

$$\Big\downarrow \text{append } Q(0)\begin{pmatrix} a_3 & a_2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x & \frac{1}{a_3} \\ 1 & 0 \end{pmatrix}$$

$$Q(a_4x^2 + a_3x + a_2) \xrightarrow{\text{Lemma 2}} Q((a_4x^2 + a_3x + a_2)x^2)$$

$$\Big\downarrow \text{append } Q(0)\begin{pmatrix} a_1 & a_0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x & \frac{1}{a_1} \\ 1 & 0 \end{pmatrix}$$

$$Q((a_4x^2 + a_3x + a_2)x^2 + a_1x + a_0).$$

Powering Efficiently

Powering Efficiently

$$\begin{array}{c}\text{Seq. of } N \text{ matrices that} \\ \text{approximately computes } Q(f)\end{array} \overset{?}{\Longrightarrow} \begin{array}{c}\text{Seq. of } \mathcal{O}(rN) \text{ matrices that} \\ \text{approximately computes } Q(f^r)\end{array}$$

Powering Efficiently

$$\text{Seq. of } N \text{ matrices that approximately computes } Q(f) \overset{?}{\Longrightarrow} \text{Seq. of } \mathcal{O}(rN) \text{ matrices that approximately computes } Q(f^r)$$

$$Q(1) \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^2) \xrightarrow[2N+4]{\text{Lemma 2}} \cdots \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^{r-2}) \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^r)$$

Powering Efficiently

Seq. of N matrices that approximately computes $Q(f)$	$\overset{?}{\Longrightarrow}$	Seq. of $\mathcal{O}(rN)$ matrices that approximately computes $Q(f^r)$

$$Q(1) \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^2) \xrightarrow[2N+4]{\text{Lemma 2}} \cdots \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^{r-2}) \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^r)$$

$$Q(f) \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^3) \xrightarrow[2N+4]{\text{Lemma 2}} \cdots \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^{r-2}) \xrightarrow[2N+4]{\text{Lemma 2}} Q(f^r)$$

Note that

- $Q(f), Q(g) \implies Q(f+g)$?
  Easy!

Note that

- $Q(f), Q(g) \implies Q(f+g)$?
  Easy!
- $Q(f) \implies Q(f^n)$?
  Easy!

Note that

- $Q(f), Q(g) \implies Q(f+g)$?
  Easy!
- $Q(f) \implies Q(f^n)$?
  Easy!
- Why is powering interesting?
  Can this compute non-sparse polynomials?

Note that

- $Q(f), Q(g) \implies Q(f+g)$?
  Easy!
- $Q(f) \implies Q(f^n)$?
  Easy!
- Why is powering interesting?
  Can this compute non-sparse polynomials?
  Yes!

Observation: There is an n-variate polynomial over $\mathbb{F}$ with $2^{\Omega(n)}$ monomials that can be approximately computed using a sequence of $n^{\mathcal{O}(1)}$ matrices, when $\mathrm{char}(\mathbb{F}) = 2$.

Conclusion and Open Problem

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?
- Can we compute $Q(fx)$ using a single $Q(f)$?

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?
- Can we compute $Q(fx)$ using a single $Q(f)$?

  Can we compute $Q(fg)$ using constant many $Q(f)$'s and $Q(g)$'s?

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?
- Can we compute $Q(fx)$ using a single $Q(f)$?

  Can we compute $Q(fg)$ using constant many $Q(f)$'s and $Q(g)$'s?

  Can we show some lower bound for $Q(f \cdot g)$?

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?
- Can we compute $Q(fx)$ using a single $Q(f)$?

  Can we compute $Q(fg)$ using constant many $Q(f)$'s and $Q(g)$'s?

  Can we show some lower bound for $Q(f \cdot g)$?
- Can we construct small size width-two ABPs for sparse polynomials?

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?
- Can we compute $Q(fx)$ using a single $Q(f)$?

  Can we compute $Q(fg)$ using constant many $Q(f)$'s and $Q(g)$'s?

  Can we show some lower bound for $Q(f \cdot g)$?
- Can we construct small size width-two ABPs for sparse polynomials?
- Is $\mathrm{VF} \subset \overline{\mathrm{VBP}_2}$ when $\mathrm{char}(\mathbb{F}) = 2$?

Conclusion and Open Problem

- The model (i.e. width-2 ABPs with approximation) is *Universal*.

Open Problems

- Is $Q(x_1 x_2 \cdots x_n)$ computable efficiently?
- Can we compute $Q(fx)$ using a single $Q(f)$?

  Can we compute $Q(fg)$ using constant many $Q(f)$'s and $Q(g)$'s?

  Can we show some lower bound for $Q(f \cdot g)$?
- Can we construct small size width-two ABPs for sparse polynomials?
- Is $\mathrm{VF} \subset \overline{\mathrm{VBP}_2}$ when $\mathrm{char}(\mathbb{F}) = 2$?

Thank you!