Non-commutative computations: lower bounds and polynomial identity testing

Guillaume Malod
Joint work with G. Lagarde, S. Perifel

IMSc Workshop on Arithmetic Complexity

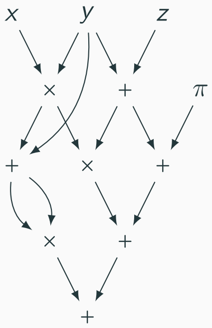
March 1st, 2017

Table of contents

1. Introduction
2. Nisan's results
3. Unambiguous circuits
4. Other results

Introduction

Arithmetic circuits



Non-commutative circuits

- $\mathbb{F}$ commutative field.
- **Non-commutative** : $xy \neq yx \rightarrow$ distinguish left and right arguments in a computation gate.
- Various motivations

Some results

- ☹ No better lower bound for NC circuits than for commutative circuits

Some results

- ☹ No better lower bound for NC circuits than for commutative circuits

  But for ABPs (Algebraic Branching Programs) :
- ☺ (Nisan 1991) Exact characterization of complexity
- ☺ (Nisan 1991) Exponential lower bounds for the *permanent*
- ☺ (Arvind, Joglekar, Srinivasan 2009) Deterministic poly-time PIT

Some results

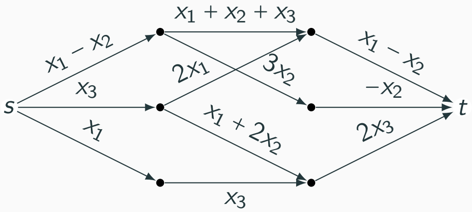
- ☹ No better lower bound for NC circuits than for commutative circuits

  But for ABPs (Algebraic Branching Programs) :
- ☺ (Nisan 1991) Exact characterization of complexity
- ☺ (Nisan 1991) Exponential lower bounds for the *permanent*
- ☺ (Arvind, Joglekar, Srinivasan 2009) Deterministic poly-time PIT

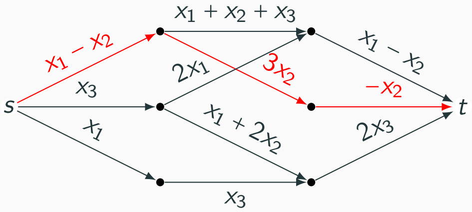
  Also (Limaye,Malod,Srinivasan 2016) Exponentiel lower bounds for *skew* circuits

Nisan's results

ABP (Branching programs)



ABP (Branching programs)

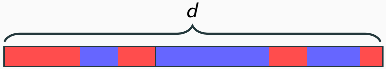


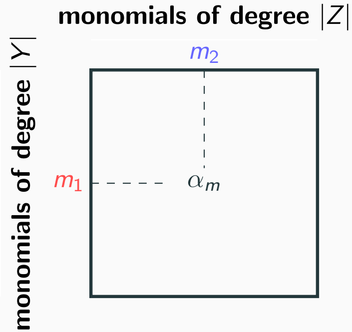
$(x_1 - x_2)(3x_2)(-x_2)$

- **DAG** : source s, sink t, edges with linear forms
- **Weight of a path** : product of edge weights
- **Computed polynomial** : sum of path weights from s to t.

Coefficient matrices

- $\Pi = ({\color{red}Y}, {\color{blue}Z})$ partition of $[d]$





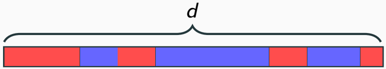
- $f = \sum_m \alpha_m . m$, homogeneous, degree d, n variables

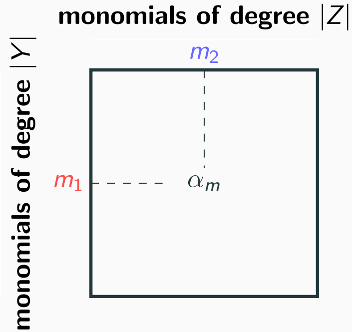
- Define matrix $M^{\Pi}(f)$

$$m \ t.q \begin{cases} m_{|Y} = m_1 \\ m_{|Z} = m_2 \end{cases}$$

Coefficient matrices

- $\Pi = (Y, Z)$ partition of $[d]$





- $f = \sum_m \alpha_m . m$, homogeneous, degree d, n variables

$$m \ t.q \begin{cases} m_{|Y} = m_1 \\ m_{|Z} = m_2 \end{cases}$$

- Define matrix $M^{\Pi}(f)$

- Complexity measure : $\text{rank}(M^{\Pi}(f))$.

Exercise: the palindrome polynomial

$$w = (w_1, \ldots, w_{d/2}) \in [n]^{d/2} \longrightarrow w^R = (w_{d/2}, \ldots, w_1)$$

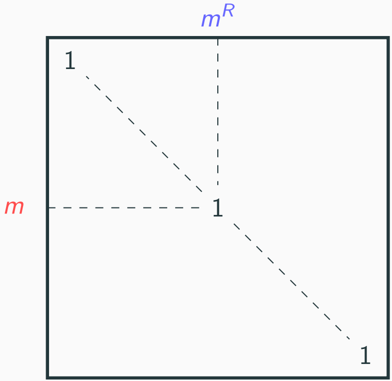
$$\tilde{x}_w = x_{w_1} x_{w_2} \ldots x_{w_{d/2}}$$

$$\mathrm{Pal}_d\, X = \sum_{w \in [n]^{d/2}} \tilde{x}_w \cdot \tilde{x}_{w^R}$$

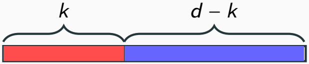
$$\mathrm{Pal}_{d+1}\, X = \sum_{i=1}^{n} x_i \cdot \mathrm{Pal}_d\, X \cdot x_i$$

What is the matrix if we cut in the middle?

Exercise: the palindrome polynomial



- $\Pi_i = (\{1, 2, \ldots, k\}, \{k+1, k+2, \ldots, d\})$

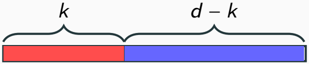


Theorem (Nisan, 1991)

For any homogeneous polynomial f of degree d, the size of a smallest homogeneous algebraic branching program for f is equal to

$$\sum_{k=0}^{d} \operatorname{rank}(M_k(f))$$

- $\Pi_i = (\{1, 2, \ldots, k\}, \{k+1, k+2, \ldots, d\})$



Theorem (Nisan, 1991)

For any homogeneous polynomial f of degree d, the size of a smallest homogeneous algebraic branching program for f is equal to

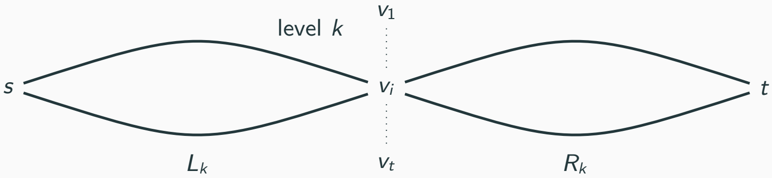
$$\sum_{k=0}^{d} \mathrm{rank}(M_k(f))$$

Corollary

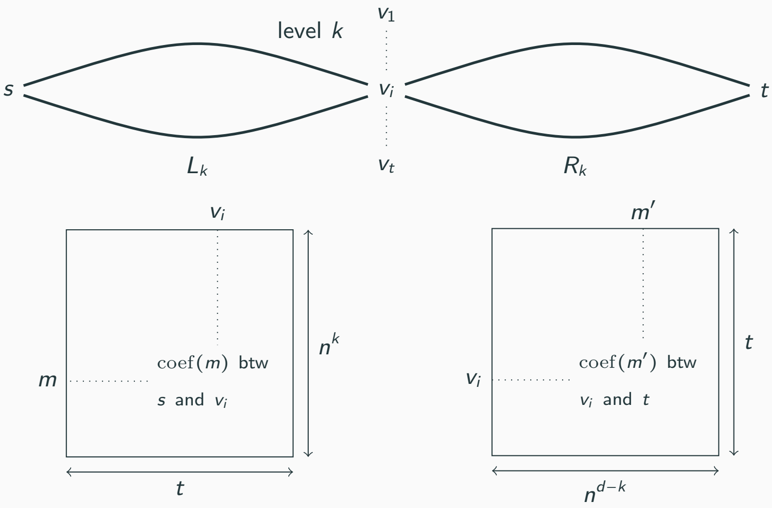
Any homogeneous ABP computing the palindrome of degree d over n variables has size $\geq n^{d/2}$

Any homogeneous ABP computing the permanent has size $\geq 2^n$

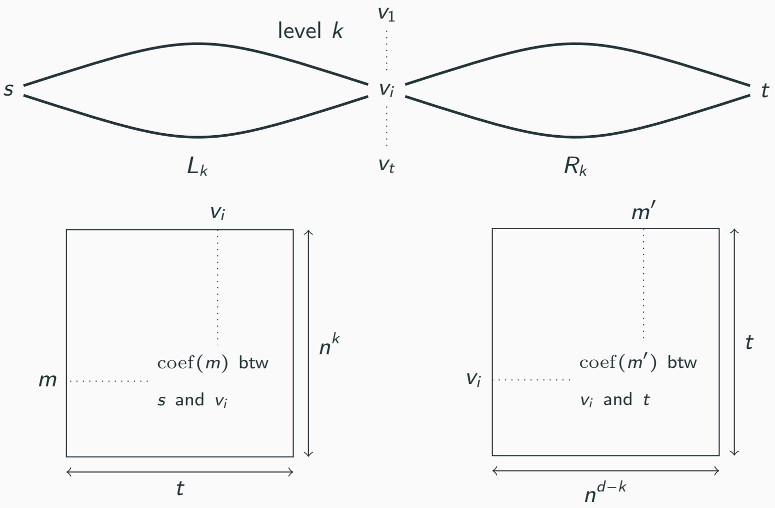
Proof (lower bound)



Proof (lower bound)

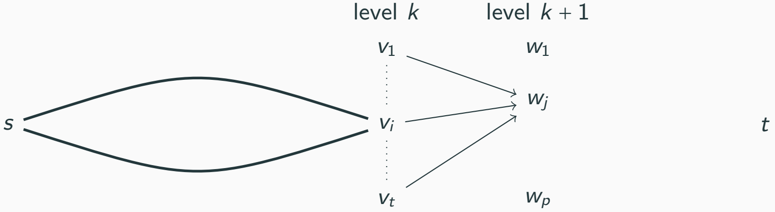


Proof (lower bound)



$$M_k(f) = L_k R_k \quad \text{and} \quad \text{rank}(M_k(f)) \leq t$$

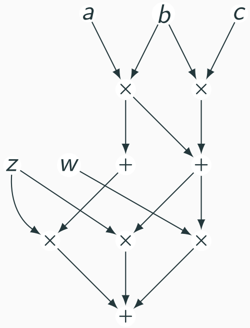
Proof (upper bound)



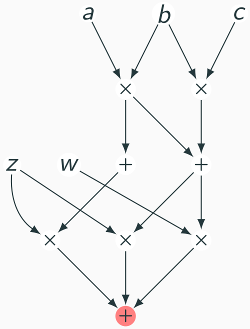
- suppose $\mathrm{rank}(L_k) < t$, then there is a column i which is a linear combination of the others
- the polynomial computed between s and v_i is a linear combination of the polynomials computed by the other vertices $v_j's$
- we could delete v_i and update the weights from level k to level $k+1$.
- so $\mathrm{rank}(L_k) = \mathrm{rank}(R_k) = t = \mathrm{rank}(M_k(f))$.

Unambiguous circuits

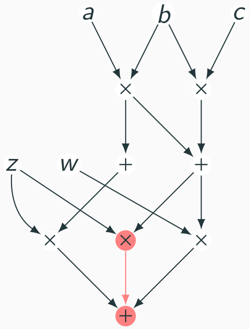
Parse trees



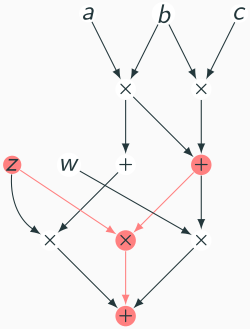
Parse trees



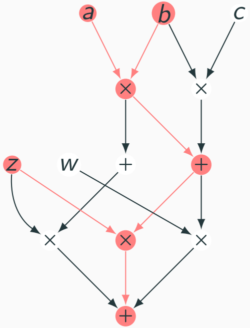
Parse trees



Parse trees



Parse trees



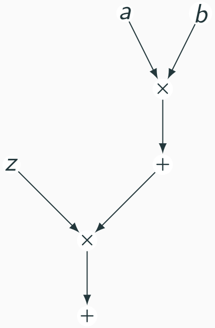


Figure 1: $val(T) = zab$

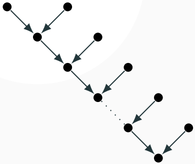
- Each parse tree computes a monomial.

Lemma

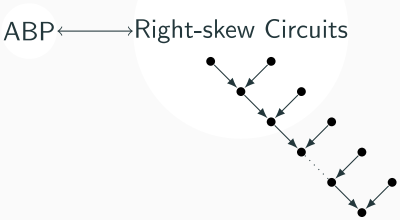
$f = \sum_{T} val(T)$

Parse trees of ABPs

ABP $\longleftrightarrow$ Right-skew Circuits



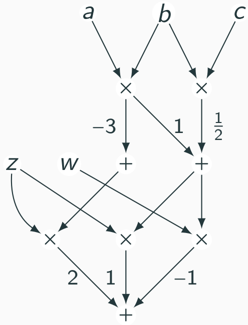
Parse trees of ABPs

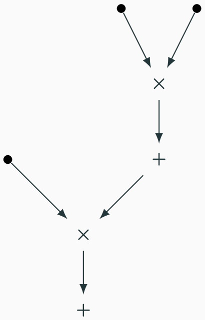


Définition

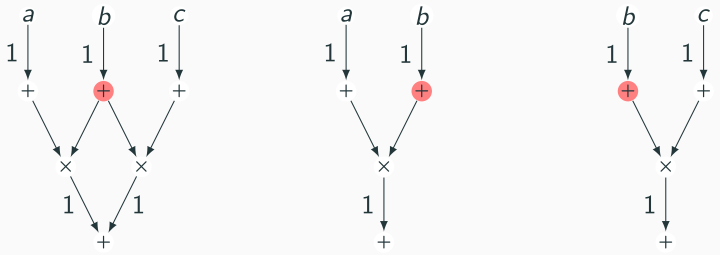
A circuit is unambiguous if all its parse trees are isomorphic.

Unambiguous circuits

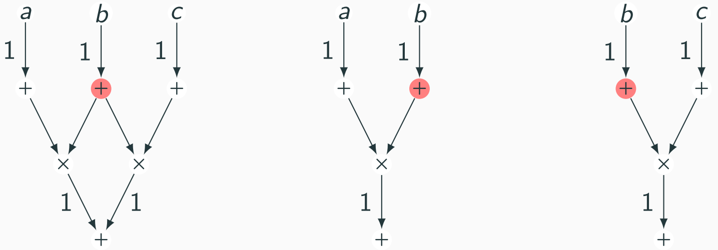




Canonical circuits

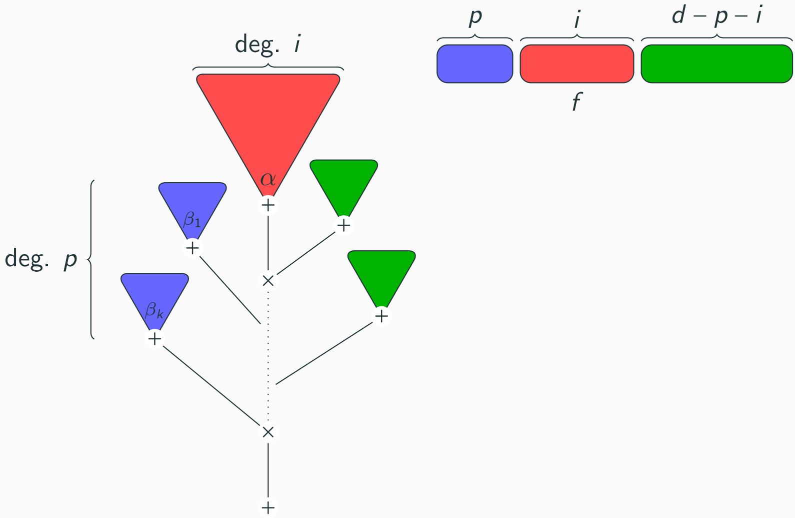


Canonical circuits



Any unambiguous circuit can be rendered canonical at a polynomial cost.

Type of a gate



Results

Theorem

Let P be a homogeneous polynomial of degree d and $\mathcal{T}$ a shape with d leaves. Then the minimal number of addition gates needed to compute P by a canonical unambiguous circuit with shape $\mathcal{T}$ is exactly equal to

$$\sum_{(i,p)\in S} \operatorname{rank}\left(M^{(i,p)}(P)\right),$$

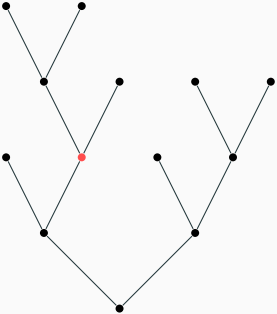
where S is the set of all existing types of +-gates in the shape $\mathcal{T}$.

Corollary

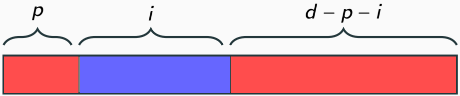
Any UC computing the permanent has size $2^{\Omega(n)}$

Proof (lower bound)

Parse tree shape



$\Pi_{(p,i)} =$



- $rg(M^{\Pi_{(p,i)}}(f)) \leq$ number of gates of type (p, i)

Proof (upper bound)

Proof (upper bound)

Clearly it works

Other results

PIT

Hadamard product (Arvind, Joglekar, Srinivasan)

Given two polynomials $P = \sum_{\vec{x}} a_{\vec{x}} \vec{x}$ and $Q = \sum_{\vec{x}} b_{\vec{x}} \vec{x}$, the *Hadamard product* of P and Q, written $P \odot Q$, equals $\sum_{\vec{x}} a_{\vec{x}} b_{\vec{x}} \vec{x}$.

Hadamard product of two unambiguous circuits

Let $\mathcal{C}$ and $\mathcal{D}$ be two unambiguous circuits in canonical form, of the same shape, and of size s and s', that compute two polynomials P and Q. Then $P \odot Q$ is computed by an unambiguous circuit of size at most ss'; moreover, this circuit can be constructed in polynomial time.

Theorem

There is a deterministic polynomial-time algorithm for PIT for polynomials computed by non-commutative unambiguous circuits over $\mathbb{R}$ (or $\mathbb{C}$).

Relationship to other classes

$ABP \subsetneq UC$

There are polynomials computed by polynomial-size UC that need exponential-size ABPs.

UC and skew are incomparable

There are polynomials computed by polynomial-size UC that need exponential-size skew circuits.
There are polynomials computed by polynomial-size skew circuits that need exponential-size UC.

The future

- Lower bounds for circuits with “similar” shapes.
- Lower bounds for circuits with not too many shapes.
- Poly-time PIT for a sum of UC circuits.

The future (much later?)

Lower bounds for general non-commutative circuits?

The future (much later?)

Lower bounds for general non-commutative circuits?

Theorem (Limaye, Malod, Srinivasan 2016)

There exists a polynomial computed by a small non-commutative circuit which is full rank for any partition.

Thank you!