Prg for tiny depth 3 suffices

- It was shown, in the last lecture, that efficient prgs for a "tiny" case of $\Sigma\wedge\Sigma\Pi$ will imply quasi-poly prg for VP.

This, of course, can also be brought down to depth 3.

→ brute-force is $s^{O(\lg s)}$.

Theorem: An efficient prg for $\Sigma\Pi\Sigma^{O(\lg s)}$ size-s circuits in $O(\lg s)$ variables over $\mathbb{F}$ $(ch\,\mathbb{F}=0) \Rightarrow n^{O(\lg n)}$-prg for VP.

Proof:

- As we have seen in the previous proof: an efficient prg gives us a multilinear polynomial family $\{q_m\}_{m\geq 1}$ that requires $\Sigma\Pi\Sigma^{O(m)}$ circuits of size $2^{\Omega(m)}$.
- As before, if $\{q_m\}_{m\geq 1}$ has a VP circuit C of size $s=2^{O(m)}$, then it can be

reduced to a $\Sigma\Lambda^{\omega(1)}\Sigma\Pi^{m/\omega(1)}$ circuit of size $2^{o(m)}$,

- which can be further reduced to a $\Sigma\Lambda^{\omega(1)}\Sigma\Lambda^{m/\omega(1)}\Sigma$ circuit of size $2^{o(m)}$, (Fischer's trick)
- which, by the duality trick on the top Λ-gate & by factorization, converts to $\Sigma\Pi\Sigma^{o(m)}$ of size $2^{o(m)}$.

$\Rightarrow$ contradiction to $\{q_m\}_m$'s hardness.

$\Rightarrow \{q_m\}_m$ requires VP circuits of size $2^{\Omega(m)}$.

$\Rightarrow$ $n^{o(\log n)}$-prg for VP. □

- Thus, all we need for PIT is to understand "tiny" depth-3 or tiny diagonal depth-4.

- How about diagonal depth-3?

Some results are known, but not completely understood.

PIT for diagonal depth-3

- Recall that the model is $C = \sum_{i=1}^{k} \ell_i^d$, where ℓ_i are linear in $\mathbb{F}[\bar{x}]$.

  (In general, $\sum_i c_i \cdot \ell_i^{d_i}$ could be resolved using similar methods.)

- The duality trick converts $(z_1 + \dots + z_s)^b$ to $\sum_{i \in [s^b (b+1)]} c_i \cdot f_i(z_1) \cdots f_i(z_s)$,

  where f_i is a deg-b polynomial.

- This can be rewritten as a vector (or matrix) product:

$$(c_1\ c_2\ c_3\ \dots) \cdot \left\{ \begin{pmatrix} f_1(z_1) \\ f_2(z_1) \\ \vdots \end{pmatrix} \cdot \begin{pmatrix} f_1(z_2) \\ f_2(z_2) \\ \vdots \end{pmatrix} \cdots \begin{pmatrix} f_1(z_s) \\ f_2(z_s) \\ \vdots \end{pmatrix} \right\}$$

- This motivates a more general model to study — read-once oblivious ABP.

ROABP

- An ABP $C(\bar{x}) = \bar{c}^T \cdot \prod_{i=1}^{h} A_i(x_i) \cdot \bar{d}$, where $\bar{c}, \bar{d}$ are $w \times 1$ & A_i are $w \times w$ matrices, over $\mathbb{F}$, is called a read-once oblivious arithmetic branching program (ROABP).

A_i is a matrix polynomial

("Oblivious" refers to the fact that the variable order is fixed in every path in the ABP.)

- ROABPs have many interesting examples:

1) Diagonal depth-3 reduces to an ROABP where the matrix product is commutative. (called commutative ROABP)

2) Multilinear depth-3 reduces to a sum of ROABPs.

A multilinear depth-3 circuit is

$$C(\bar{x}) = \sum_{i\in[k]} \prod_{j\in[d]} \ell_{ij}, \text{ with } \{\ell_{ij} \mid j\}_i$$

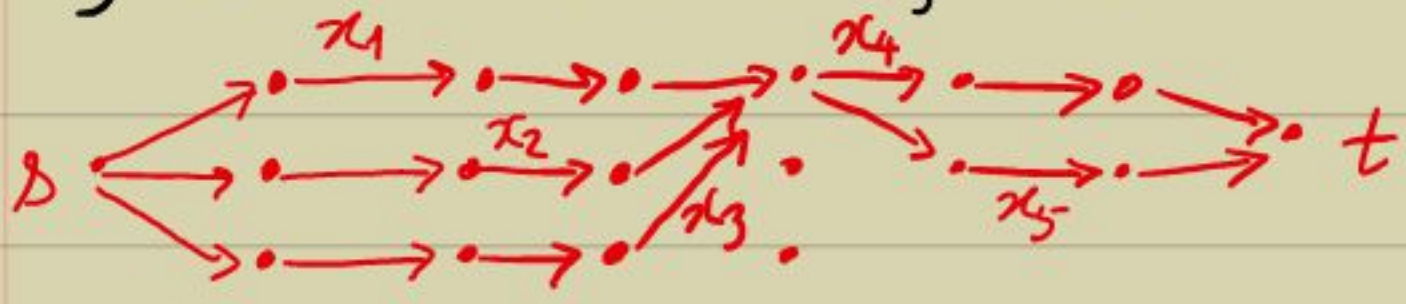
for any i, being linear polynomials on disjoint variables.

$\Rightarrow$ the i-th product gate induces a partition P_i on $[n]$.

▷ For a partition $[n] = S_1 \sqcup \cdots \sqcup S_d$, any product $\prod_{i\in[d]} \ell_i(x_{S_i})$ is a width-n ROABP.

Pf$_{\Rightarrow i}$

• Eg. $(x_1+x_2+x_3)(x_4+x_5)$ can be computed by the ROABP diagram:



□

▷ $\sum_{i\in[k]} \prod_{j\in[d]} \ell_{ij}(x_{S_j})$ has a width-(kn) ROABP.

Pf: • Use k copies of the above diagram, in parallel.

□

- The above model is called set-multilinear $\Sigma^k\Pi^d\Sigma$ circuit.

- It is now clear that:

▷ A set-multilinear $\Sigma\Pi\Sigma$ is an ROABP.

▷ $\Sigma\wedge\Sigma$ is a commutative ROABP.

▷ Multilinear $\Sigma\Pi\Sigma$ is a sum of ROABPs.

Whitebox PIT for ROABP

- This is completely solved!

Thm [Raz-Shpilka '05] The width-w, individual-deg-d, n-variate ROABP has a whitebox poly(wdn)-time PIT algorithm.

Proof: