Outline

1. (Long) Introduction
2. Randomized Polynomials (w/applications to round-efficient MPC)
3. Randomized Encodings w/applications to NC^0 Cryptography
4. Constant Input Locality
5. Computational Randomized Encodings (w/applications)
6. NC^0 Linear Stretch PRG (w/applications)

Polynomial Representations

- Fix a finite field F.
- Standard representation:

$p \in F[x_1,\ldots,x_n]$ represents $f:\{0,1\}^n \to \{0,1\}$ if

$$\forall\, x \in \{0,1\}^n \quad p(x)=f(x)$$

- Applications in:

  circuit lower-bounds

  interactive proofs and PCP

  worst-case to average-case reductions

  ....

- Main complexity measure: degree.

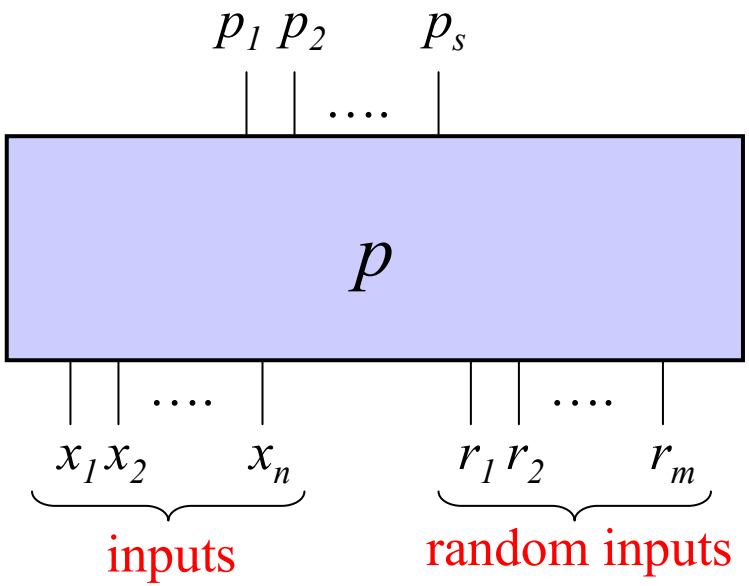
$$\deg(x_2x_3^8 + x_1x_2x_3) = 9$$

- Degree n is sufficient for all $f: \{0,1\}^n \to \{0,1\}$.

  Example. $x_1 \vee x_2$ by $(1-x_1)x_2+x_1(1-x_2)+x_1x_2 = x_1+x_2- x_1x_2$

- Degree n is necessary for most f.

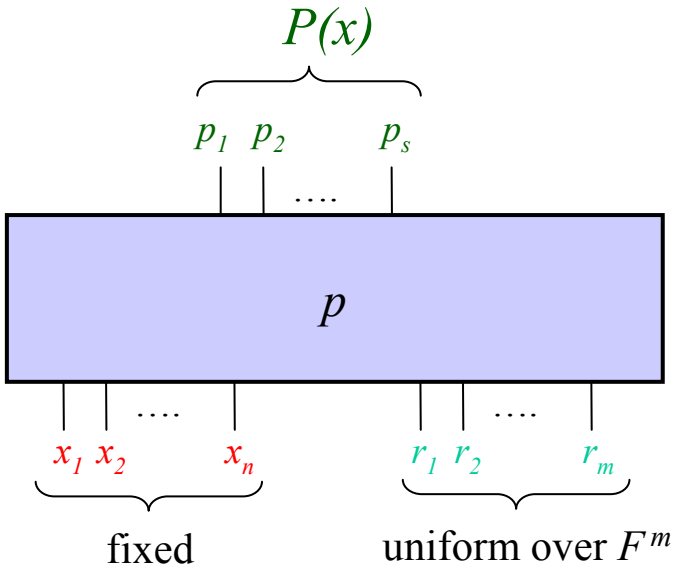
Randomizing Polynomials - Syntax



- $p=(p_1(x,r),p_2(x,r),\ldots,p_s(x,r))$
- $\deg(p)= \max \{ \deg(p_i) \}$
- output complexity = s
- randomness complexity = m

Randomizing Polynomials - Semantics

- *P(x) - output distribution* on input *x*.



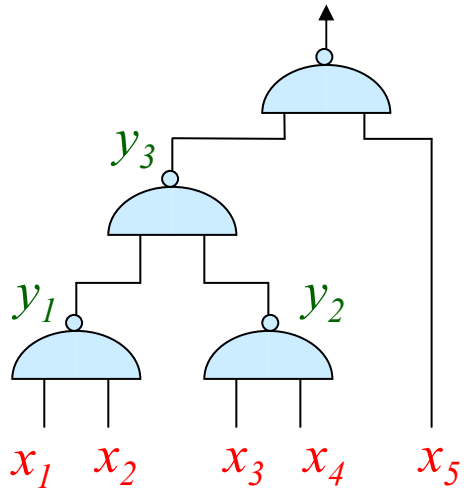
representing $f:\{0,1\}^n \rightarrow \{0,1\}$ by…	requirements
standard polynomial *p(x)*	$f(x)=0 \Rightarrow p(x)=0$ $f(x)=1 \Rightarrow p(x)=1$
randomizing polynomials *p(x,r)*	$f(x)=0 \Rightarrow P(x) \equiv D_0$ $f(x)=1 \Rightarrow P(x) \equiv D_1$ } privacy $dist(D_0, D_1) \geq 1/2$ } correctness

Examples

function	field	representation	degree	output complexity	randomness complexity
XOR	GF(2)	$x_1+x_2+\ldots+x_n$	1	1	0
OR	*any*	$x_1r_1+x_2r_2+\ldots+x_nr_n$	2	1	n
AND	*any*	$(1-x_1)r_1+\ldots+(1-x_n)r_n$	2	1	n
MAJ	?	?	?	?	?
any	*any*		3	$O(\text{BP-size}^2)$	

3 Ways to Degree 3

1. Degree-3 construction using circuit representation



$$f(x)=1 \quad \Leftrightarrow \quad \begin{aligned} &\exists\, y_1, y_2, y_3 \\ &y_1=\mathrm{NAND}(x_1,x_2)= x_1(1-x_2)+(1-x_1)x_2+(1-x_1)(1-x_2) \\ &y_2=\mathrm{NAND}(x_3,x_4) \\ &y_3=\mathrm{NAND}(y_1,y_2) \\ &1 =\mathrm{NAND}(y_3,x_5) \end{aligned}$$

Note: $\Rightarrow$ $\exists!\, y_1, y_2, y_3$

Using circuit representation (contd.)

$$\left.\begin{array}{l} q_1(x,y)=0 \\ q_2(x,y)=0 \\ \quad\ldots \\ q_k(x,y)=0 \end{array}\right\} \text{ deg.-2}$$

$$p(x,\, y,r)=\Sigma\, r_i\, q_i(x,y) \quad \Big\} \text{ deg.-3}$$

$f(x)=0 \;\Rightarrow\;$ $P(x)$ is uniform

$f(x)=1 \;\Rightarrow\;$ $P(x) \equiv 0$ given $y=y_0$, otherwise it is uniform

Statistical distance amplified to 1/2 by $2^{\Theta(k)}$ repetitions.

- complexity exponential in circuit size
- works over any field

2. Degree-3 construction using quadratic characters

- QR_t = subgroup of quadratic residues in Z_t^*
- $\chi_t(i) = \begin{cases} 0, & i \in QR_t \\ 1, & i \notin QR_t \end{cases}$

Fact from number theory:

$$\forall N \ \ \forall \text{bit-sequence } b \in \{0,1\}^N$$

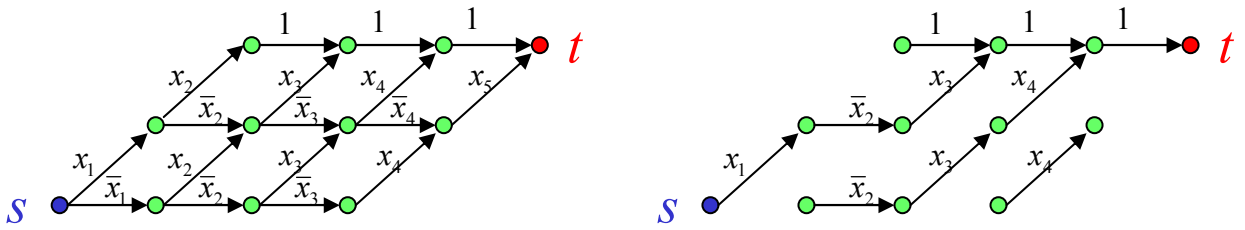
$$\exists \text{prime } t(=2^{O(N)}) \ \ \exists d > 0 \ \text{ such that } b = \chi_t(d)\chi_t(d+1)\cdots\chi_t(d+N-1)$$

- Let $N=2^n$, b = length-N truth-table of f, F=GF(t)
- Define $p(x_1,\ldots,x_n,r) = \left(d + \sum_{i=1}^{n} 2^{i-1} x_i\right) \cdot r^2$

  - one polynomial
  - huge field size

Degree-3 Construction from Branching Program Representation

BP=(G, s , t, edge-labeling) G_x=subgraph induced by x



mod-q Branching Program: accepts x if # s-t paths in $G_x \not\equiv 0 \pmod{q}$

- size = # of vertices
- polynomial-size branching programs capture log-space classes
- many natural problems admit linear-size branching programs

Lemma [IK97]: Let F=GF(q), where q is prime.

Suppose that f has a mod-q branching program of size a.

Then there is a degree-1 mapping $L:F^n \to F^{a,a}$ such that:

- $f(x)=1 \Rightarrow$ rank($L(x)$)=a
- $f(x)=0 \Rightarrow$ rank($L(x)$)=a-1

Degree 3 construction based on Lemma: $p(x,R_1,R_2)= R_1L(x)R_2$

•Privacy:

rank(M)=rank(M') $\Rightarrow$ $M = Q_1M'Q_2$ for some invertible Q_1,Q_2

$\Rightarrow$ $R_1MR_2 \equiv R_1(Q_1M'Q_2)R_2 \equiv (R_1Q_1)M'(Q_2R_2) \equiv R_1M'R_2$

•Correctness:

Pr[both R_1 and R_2 are of full rank] > 0.08

Stat. distance amplified to 1/2 by $O(1)$ repetitions.

•Complexity: $O(a^2)$

Proof of Lemma

- Given: mod-q branching program of size $a+1$
- Let A_x be the $(a+1)\times(a+1)$ adjacency matrix of G_x over $F=\mathrm{GF}(q)$
- Let M_x denote the $a\times a$ (t,s)-minor of $I-A_x$

$$A_x = \begin{array}{c} \\ s \\ v_1 \\ v_2 \\ v_3 \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} s & v_1 & v_2 & v_3 & t \end{array} \\ \begin{bmatrix} 0 & x_1 & 1-x_1 & 0 & 0 \\ 0 & 0 & x_2 & 1 & 0 \\ 0 & 0 & 0 & 1-x_5 & x_4 \\ 0 & 0 & 0 & 0 & x_3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad M_x = \begin{array}{c} \\ s \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & t \end{array} \\ \begin{bmatrix} -x_1 & x_1-1 & 0 & 0 \\ 1 & -x_2 & -1 & 0 \\ 0 & 1 & x_5-1 & -x_4 \\ 0 & 0 & 1 & -x_3 \end{bmatrix} \end{array}$$

- $\#s\text{-}t$ paths in $G_x \bmod q = (I + A_x + A_x^2 + \ldots)_{s,t} = (I - A_x)^{-1}_{s,t} = \dfrac{\det(M_x)}{\det(I - A_x)}$

$$\#s\text{-}t \text{ paths} \not\equiv 0 \;\Rightarrow\; \det(M_x) \neq 0 \;\Rightarrow\; \mathrm{rank}(M_x) = a$$

$$\#s\text{-}t \text{ paths} \equiv 0 \;\Rightarrow\; \det(M_x) = 0 \;\Rightarrow\; \mathrm{rank}(M_x) = a\text{-}1$$

- Letting $L(x)=M_x$, all requirements are satisfied.

Application of Randomizing Polynomials to Secure Computation

Motivation: round complexity

Generic protocols [BGW88,CCD88,GMW87,RB89,CDM00,...] :

- require many rounds for computing general functions.
- require few rounds for low-degree polynomials.

  Degree-3: 2 rounds with $t<k/3$, 3 rounds with $t<k/2$.

The secure computation of an arbitrary function f can be reduced to the secure computation of degree-3 polynomials.

Proof outline:

- Let $p(x,r)$ be degree-3 randomizing polynomials for f.
- Goal: securely sample from $P(x)$.
- Solution: securely compute $p'(x,r^1,r^2,\ldots,r^{t+1})=p(x, r^1+\ldots+r^{t+1})$
  where each r^j is randomly picked by a different party.

Note: $\deg(p')=\deg(p)$

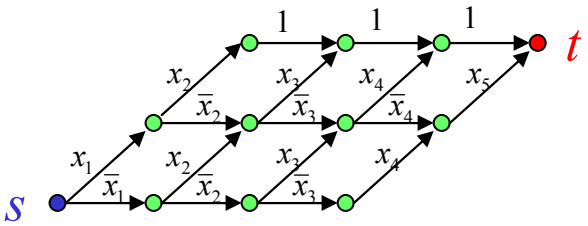
Corollary: 2 rounds (3 rounds) suffice for k parties to t-privately compute any function f with $t<k/3$ ($t<k/2$), probabilistic correctness, and quadratic communication in the branching program size of f.

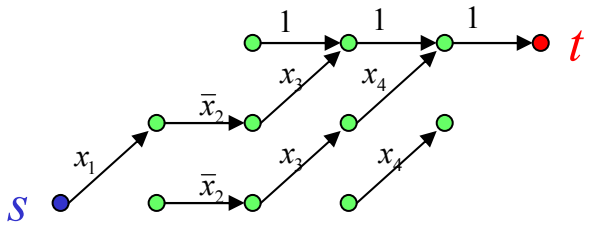
Is Perfect Correctness Possible?

- In previous constructions, output distributions overlap.
  - Inevitable when each monomial contains r_i
- Constant-round i.t. protocols from the literature either make errors or use expected constant rounds.
- Goal: obtain perfect constant-round protocols via perfect randomizing polynomials.

Perfect Degree-3 Encoding of BPs

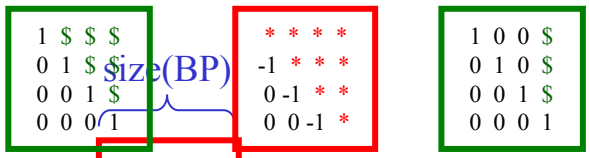
BP=(G, s, t, edge-labeling) G_x=subgraph induced by x



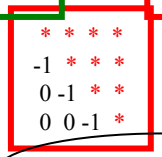


Encoding based on Lemma: $g(x,r_1,r_2) = R_1(r_1)\cdot L(x)\cdot R_2(r_2)$

mod-q CBP: $f(x)$ = # $s\to t$ paths in G_x mod q.

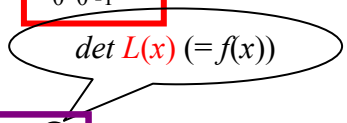


Lemma: $\exists$ degree-1 mapping $L: x \to$

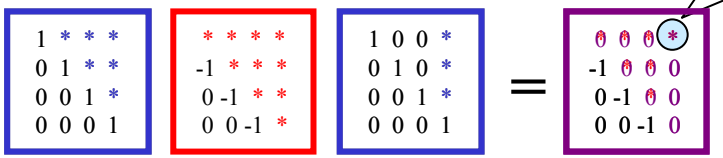


s.t. $\det(L(x)) = f(x)$.

Correctness: $f(x)$=*det* $g(x,r_1,r_2)$



Privacy:



$g(x,r_1,r_2) \equiv$