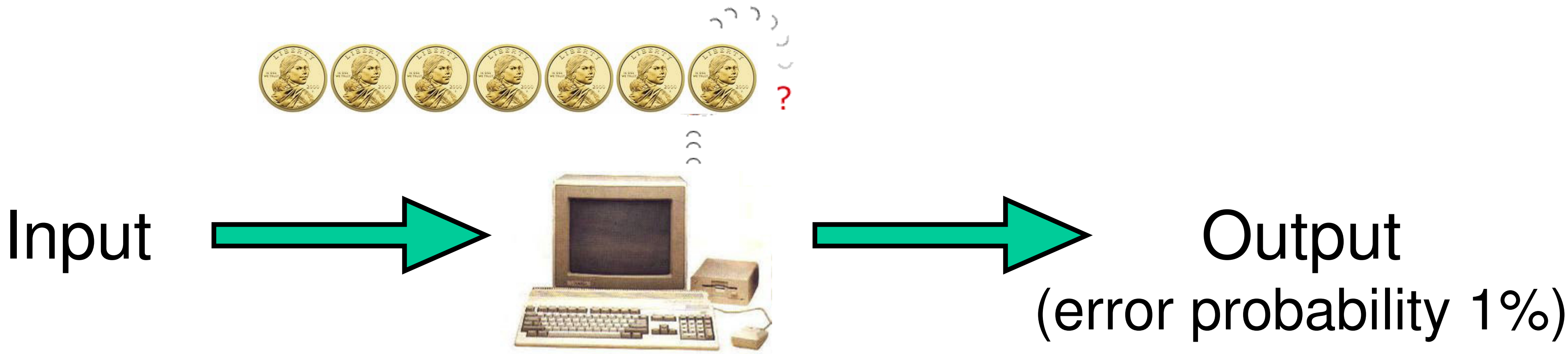
Pseudorandomness: New Results and Applications

Emanuele Viola
IAS

May 2007

Randomness in Computation



- Useful throughout Computer Science
  - Algorithms
  - Cryptography
  - Complexity Theory

- Question: Is “true” randomness necessary?

Pseudorandomness

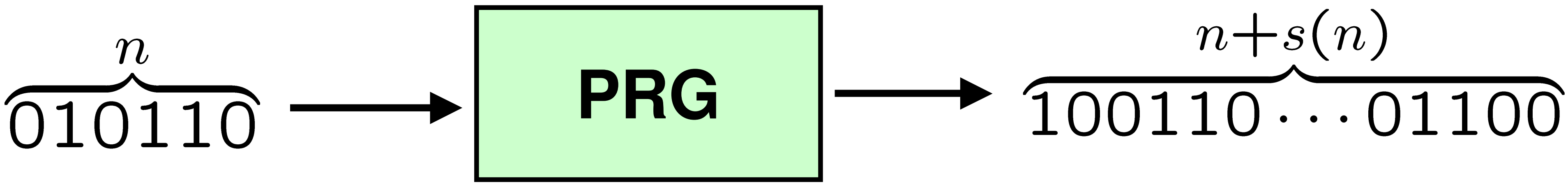
- Goal: low-entropy distributions
  that ``look random''

- Why study pseudorandomness?

- Basis for most cryptography [S 49]

- Algorithmic breakthroughs:
  Connectivity in logarithmic space [R 04]
  Primality in polynomial time [AKS 02]

Pseudorandom Generator (PRG) [BM,Y]

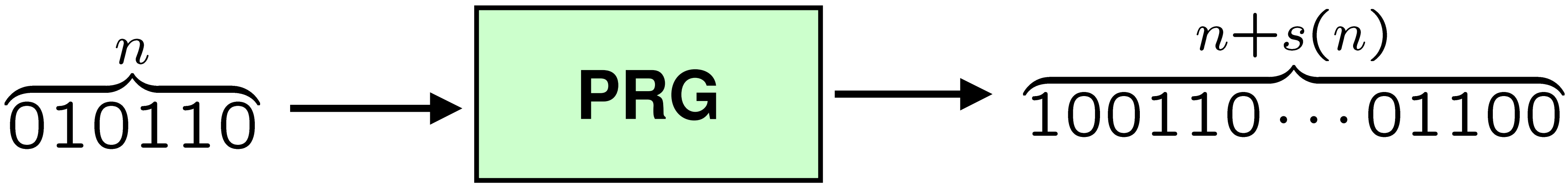


- Poly(n)-time Computable
- Stretch $s(n) \geq 1$ (e.g., $s(n) = 1$, $s(n) = n^2$)
- Output ``looks random''

Outline

- Overview of pseudorandomness
- Cryptographic pseudorandom generators
  - Complexity vs. stretch
- Specialized pseudorandom generators
  - Constant-depth, with application to NP
  - Polynomials

Cryptographic PRG



- "Looks random": ∀ efficient adversary A : $\{0,1\}^{n+s(n)} \to \{0,1\}$

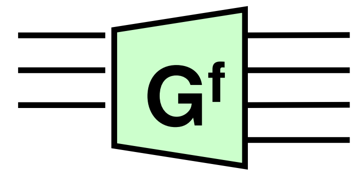
$$\Pr_U[A(U) = 1] \approx \Pr_X[A(PRG(X)) = 1]$$

- Cryptography: sym. encryption(m) := m $\oplus$ G(X) [S49]
  need big stretch s >> n

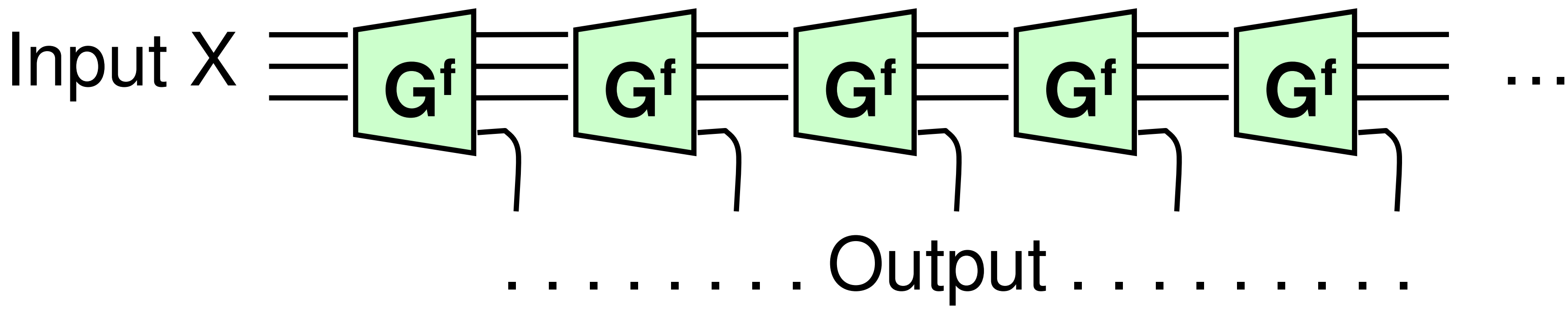
- PRG $\Leftrightarrow$ One-Way Functions (OWF) [BM,Y,GL,…,HILL]
  - OWF: easy to compute but hard to invert

Standard Constructions w/ big stretch

- STEP 1: OWF f ⇒ $G^f : \{0,1\}^n \rightarrow \{0,1\}^{n+1}$
  - Think e.g. $f : \{0,1\}^{n^a} \rightarrow \{0,1\}^{n^b}$



- STEP 2: G^f ⇒ PRG with stretch s(n) = poly(n) [GM]



- Stretch s ⇒ s adaptive queries to f ⇒ circuit depth $\geq$ s

- Question [this work]: stretch s vs. adaptivity & depth?
  E.g., can have s = n, circuit depth O(log n)?

Previous Results

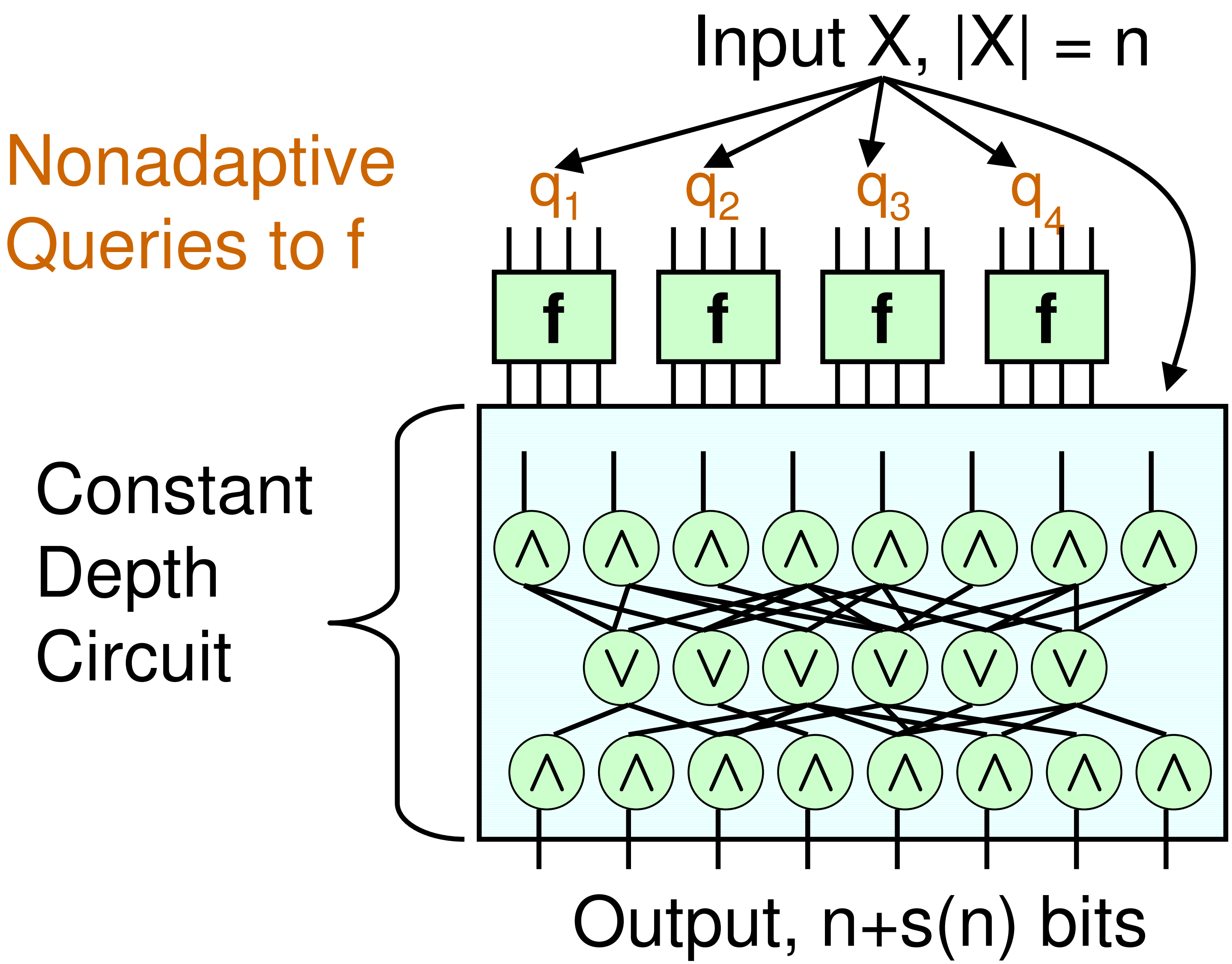
- [AIK] Log-depth OWF/PRG $\Rightarrow$ O(1)-depth PRG (!!!)
  However, any stretch $\Rightarrow$ stretch s = 1

- [GT] s vs. *number* q of queries to OWF (Thm: q $\geq$ s)
  [This work] s vs. *adaptivity & circuit depth*

- [...,IN,NR] O(1)-depth PRG from *specific* assumptions
  [We ask] *general* assumptions

Our Model of PRG construction

- **Parallel** PRG $G^f : \{0,1\}^n \to \{0,1\}^{n+s(n)}$ from OWF f



Our Results on PRG Constructions

- Theorem [V] Parallel $G^f : \{0,1\}^n \to \{0,1\}^{n+s(n)}$ from OWF (e.g. $f : \{0,1\}^{n^a} \to \{0,1\}^{n^b}$) must have:

	f arbitrary	**f one-to-one**	**f permutation**
Neg.	$s(n) \leq o(n)$	$s(n) \leq o(n)$	?
Pos.	?	$s(n) \geq 1$	$s(n) \geq 1$

Proof of positive result

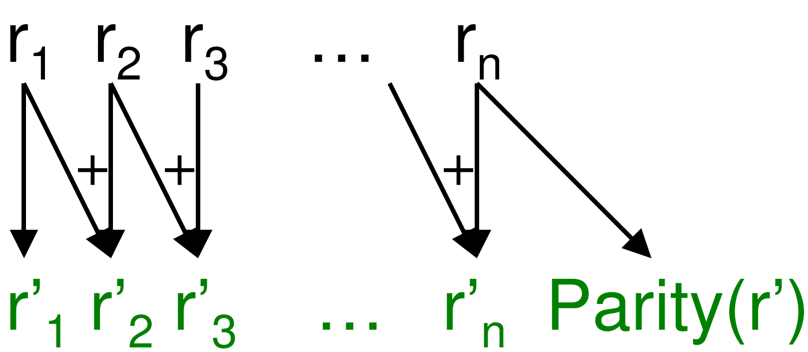
Setting: f = permutation π, want stretch s = 1

[GL] $G^f(x,r) := \pi(x),r,<x,r>$ $(<x,r> := \sum_i x_i r_i)$

Problem: can't compute $<x,r>$ in constant-depth [GNR]

Solution: don't have to! $G^f(x,r) := \pi(x),r',<x,r'>$

Easier: generate random $(r',\text{Parity}(r') := \sum_i r_i)$:

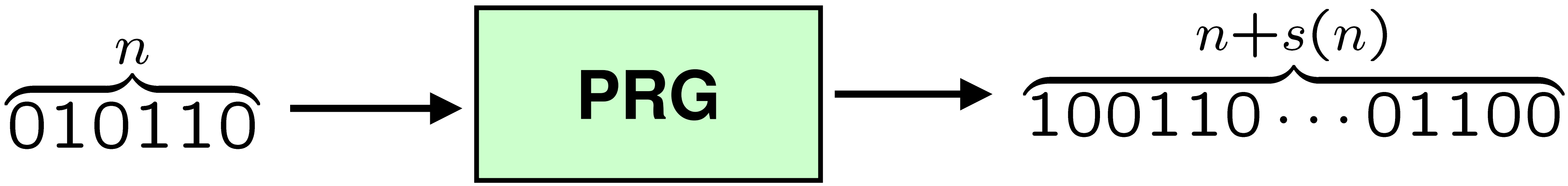


Technique gives $<x,r'>$, extractors, etc. Q.E.D.

Outline

- Overview of pseudorandomness
- Cryptographic pseudorandom generators
  - Complexity vs. stretch
- Specialized pseudorandom generators
  - Constant-depth, with application to NP
  - Polynomials

Specialized PRG



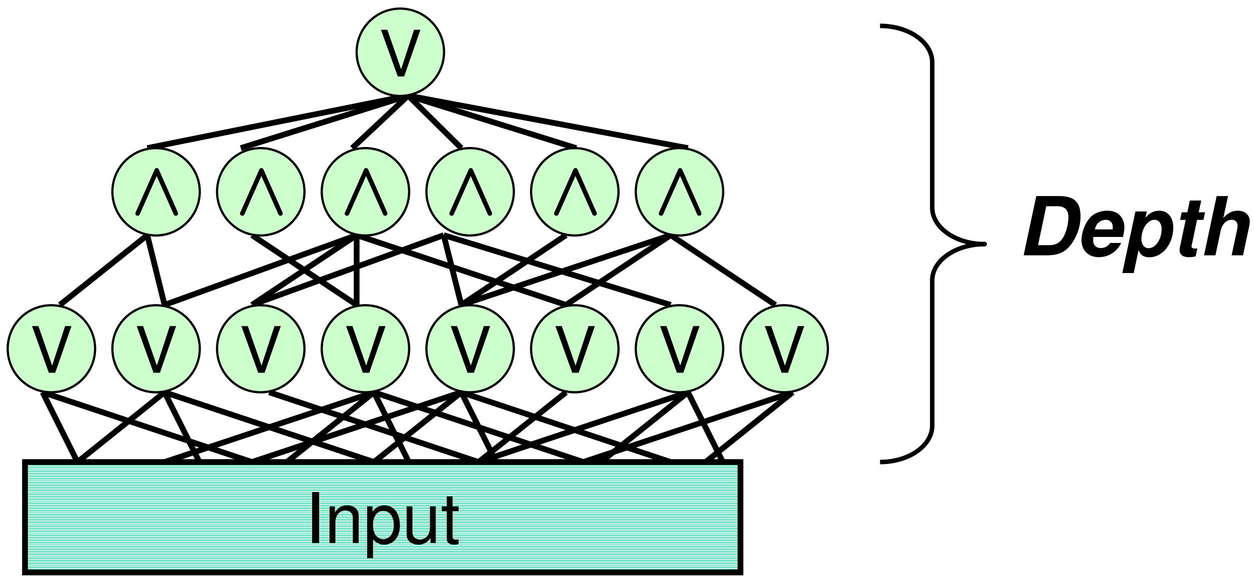
- "looks random": ∀ **restricted** A : $\{0,1\}^{n+s(n)} \rightarrow \{0,1\}$

$$\Pr_U[A(U) = 1] \approx \Pr_X[A(PRG(X)) = 1]$$

- Sometimes known unconditionally!

PRG for Constant-Depth Circuits

- Constant-depth circuit:



- Theorem [N '91]: PRG with stretch $s(n) = 2^{n^{\Omega(1)}}$
  output looks random to constant-depth circuits

Application: Avg-Case Hardness of NP

- Study hardness of NP on random instances
  - Natural question, essential for cryptography

- Currently cannot relate to P $\neq$ NP [FF,BT,V]

- Hardness amplification

Definition: $f : \{0,1\}^n \rightarrow \{0,1\}$ is ε-hard if

$\forall$ efficient algorithm M : $\Pr_x[M(x) \neq f(x)] \geq 1/2 - \varepsilon$

Previous Results

- Yao's XOR Lemma: $f'(x_1,\ldots, x_n) := f(x_1) \oplus \cdots \oplus f(x_n)$
  $f' \approx 2^{-n}$ -hard, almost optimal

- Cannot use XOR in NP: $f \in NP \not\Rightarrow f' \in NP$

- Idea: $f'(x_1,\ldots, x_n) = C(f(x_1),\ldots, f(x_n))$, C monotone
  - e.g. $f(x_1) \wedge (f(x_2) \vee f(x_3))$. $f \in NP \Rightarrow f' \in NP$

- Theorem [O'D]: There is C s.t. $f' \approx (1/n)$-hard

- Barrier: No monotone C can do better!

Our Result on Hardness Amplification

- Theorem [HVV]: Amplification in NP up to $\approx 2^{-n}$
  - Matches the XOR Lemma

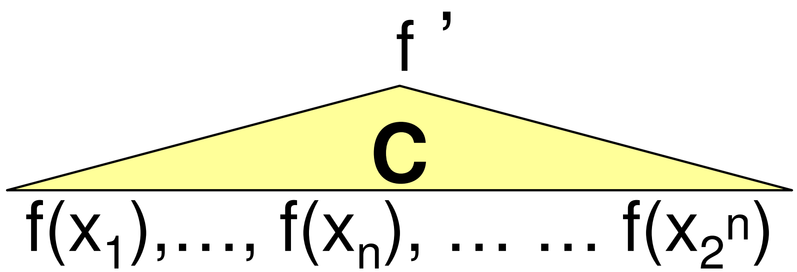
- Technique: Pseudorandomness!

  Intuitively, $f' := C(f(x_1),\ldots, f(x_n), \ldots \ldots f(x_{2^n}))$

  f' $(1/2^n)$-hard by previous result

  Problem: Input length = 2^n

  Note C is constant-depth

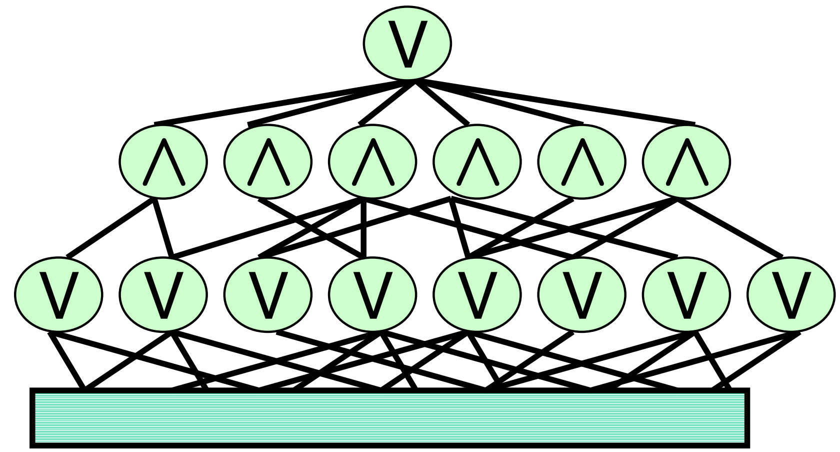


  Use PRG: input length $\rightarrow$ n, keep hardness

Previous Results

- Recall Theorem [N]:

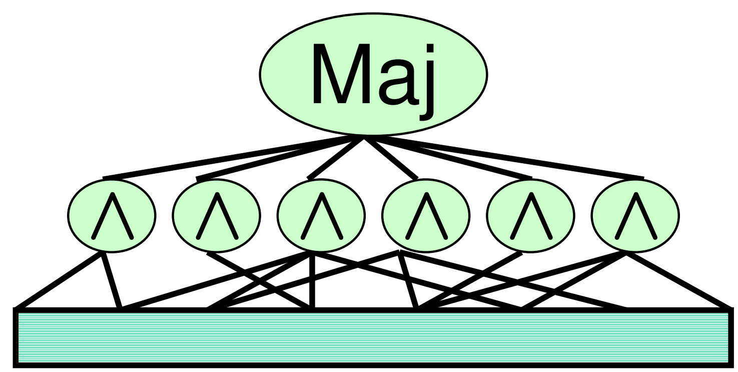
  PRG with stretch $s(n) = 2^{n^{\Omega(1)}}$



- But constant-depth circuits are weak:
  – Cannot compute $\text{Majority}(x_1,\ldots,x_n) := \sum_i x_i > n/2$?

- Theorem [LVW]:

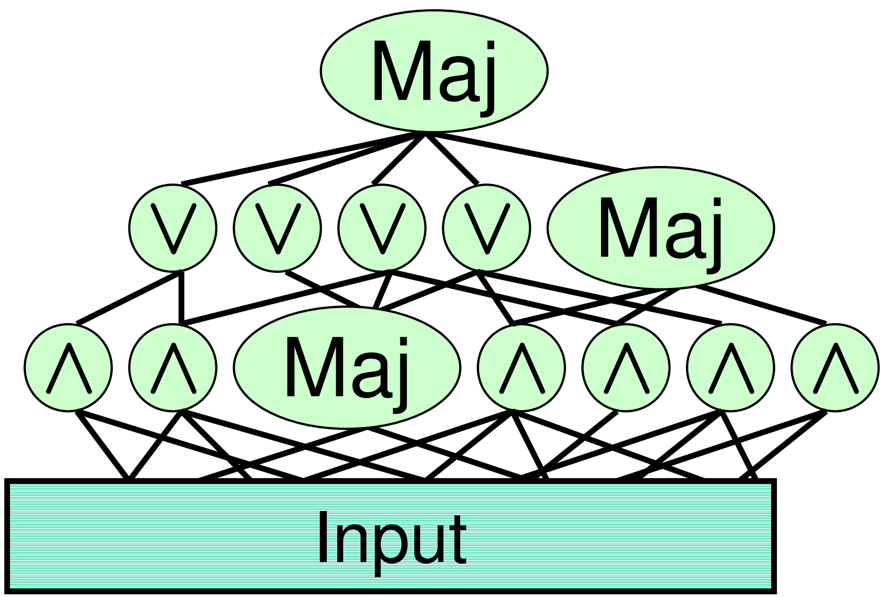
  PRG with stretch $s(n) = n^{\log n}$



- PRG's for incomparable classes

Our New PRG

- Constant-depth circuits with few Majority gates



- Theorem **[V] :** PRG with $s(n) = n^{\log n}$

- Improves on [LVW]; worse stretch than [N] Richest class for which PRG is known

- Techniques: Communication complexity + switching lemma [BNS,HG,H,HM,CH]

Outline

- Overview of pseudorandomness
- Cryptographic pseudorandom generators
  - Complexity vs. stretch
- Specialized pseudorandom generators
  - Constant-depth, with application to NP
  - Polynomials

F_2 polynomials

- Field $F_2 = GF(2) = \{0,1\}$
- F_2-polynomial $p : F_2^n \rightarrow F_2$ of degree d

  E.g., $p = x_1 + x_5 + x_7$ $\quad$ $d = 1$

  $p = x_1 \cdot x_2 + x_3$ $\quad$ $d = 2$

- Theorem[NN90]: PRG for d=1 with stretch $s(n)=2^{\Omega(n)}$
  - Applications to algorithm design, PCP's,…

Hardness for F_2 polynomials

- Want: explicit $f : \{0,1\}^n \to \{0,1\}$ ε-hard for degree d:
  $\forall$ p of degree d : $\Pr[f(x) \neq p(x)] \geq ½ - \varepsilon$
  $\varepsilon = \varepsilon(n,d)$ small

- Implies PRG with s=1. G(X) := X f(X)

- Interesting beyond PRG
  - Coding theory
  - $d = \log n$, $\varepsilon = 1/n^{10} \Rightarrow$ complexity breakthrough

Previous Results

- Want: explicit $f : \{0,1\}^n \rightarrow \{0,1\}$ ε-hard for degree d:
  $\forall$ p of degree d : $\Pr[f(x) \neq p(x)] \geq ½ - \varepsilon$
  $\varepsilon = \varepsilon(n,d)$ small

- [Razborov 1987] Majority: (1/n)-hard (d $\leq$ polylog(n))

- [Babai et al. 1992] Explicit f: $\exp(-n/d\cdot 2^d)$-hard

- [Bourgain 2005] Mod 3: $\exp(-n/8^d)$-hard
  - Mod 3 $(x_1,\ldots,x_n) := 1$ iff $3 \mid \sum_i x_i$

Our Results

- New approach based on ``Gowers uniformity''

- Theorem [V,VW] :

  Explicit f: $\exp(-n/2^d)$-hard ([BNS] $\exp(-n/d \cdot 2^d)$)

  Mod 3: $\exp(-n/4^d)$-hard ([Bou] $\exp(-n/8^d)$)

  - Also arguably simpler proof

- Theorem [BV, unpublished] :

  PRG with stretch $s(n) = 2^{\Omega(n)}$ for d = 2,3

  - For any d under "Gowers inverse conjecture"
  - Even for d=2, previous best was $s(n) = n^{\log n}$ [LVW '93]

Gowers uniformity

- Idea: Measure closeness to degree-d polynomials by checking if d-th derivative vanishes
  - [G98] combinat., [A+,J+,…] testing

- Derivative $D_y\, p(x) := p(x+y) + p(x)$
  - E.g. $D_y\,(x_1x_2 + x_3) = (y_1+x_1)(y_2+x_2)+(x_3+y_3)+x_1x_2+x_3$
    $= y_1x_2 + x_1y_2 + y_1y_2 + y_3$
  - p degree d $\Rightarrow D_y\, p(x)$ degree d-1
  - Iterate: $D_{y,y'}\, p(x) := D_y(D_{y'}\, p(x))$

- d-th Gowers uniformity of f:
  $U_d(f) := E_{x,y^1,\ldots,y^d}[e(D_{y^1,\ldots,y^d}\, f(x))]$ $\qquad (e(X):=(-1)^X)$
  - $U_d(p) = 1$ if p degree d

Main lemma

- Lemma [Gow,GT]:
  Hardness of f for degree-d polynomials $\leq U_d(f)^{1/2^d}$
  - Property of f only!

- Proof sketch: Let p have degree d.
  Hardness of f for p

  $= |\ \Pr[f(x) = p(x)] - \Pr[f(x) \neq p(x)]\ |$

  $= E_X[e(f(x)+p(x))] = U_0(f+p)$

  $\leq U_1(f+p)^{1/2} \leq \ldots \leq U_d(f+p)^{1/2^d}$ (Cauchy-Schwartz)

  $= U_d(f)^{1/2^d}$ (d-th derivative of p = 1)

Q.E.D.

Establishing hardness

- Consider $f := x_1 \cdots x_{d+1} + x_{d+2} \cdots x_{2d+2} + \cdots$
  - not best parameters, but best to illustrate

- Theorem [V] f is $\exp(-n/c^d)$-hard for degree d

- Proof:
  Hardness of f $\leq U_d(f)^{1/2^d}$ (by lemma)

  $= U_d(x_1 \cdots x_{d+1} + x_{d+2} \cdots x_{2d+2} + \cdots)^{1/2^d}$

  $= U_d(x_1 \cdots x_{d+1})^{n/(d+1)2^d}$ (by property of U)

  $= \exp(-n/c^d)$ (by calculation)

Q.E.D.

Conclusion

- Pseudorandom generators (PRG's): powerful tool

- Cryptographic PRG's
  - Tradeoff between stretch and parallel complexity [V]

- Specialized PRG's
  - Application: Hardness Amplification in NP [HVV]
  - PRG for const.-depth circuits with few Maj gates [V]
  - PRG for low-degree polynomials over F_2 using Gowers uniformity [V, VW,BV]

Thank you!