

On generating independent random strings

Marius Zimand
Towson University

CiE 2009 Heidelberg
July 23, 2009

Basic Questions

Given some randomness is it possible to effectively produce **better** randomness?

$$x \mapsto f(x)$$

$$\text{Randomness}[f(x)] > \text{Randomness}[x]$$

Basic Questions

Given some randomness is it possible to effectively produce **better** randomness?

$$x \mapsto f(x)$$

$$\text{Randomness}[f(x)] > \text{Randomness}[x]$$

Given some randomness is it possible to effectively produce **new and better** randomness?

$$x, y \mapsto f(x, y)$$

$$\text{Randomness}[f(x, y) \mid x] \text{ high}$$

$$\text{Randomness}[f(x, y) \mid y] \text{ high}$$

Randomness[x] = Kolmogorov complexity of x

Kolmogorov complexity of a string is the length of its shortest description.

- $\overbrace{0101 \dots 01}^{10^{100}}$ has a short description.
- flipping a coin 10^{100} times:
011000101010110010101000101001011...100: description $\approx 10^{100}$ bits.

Plain Kolmogorov complexity

$$C(x) = \min\{|p| \mid U(p) = x\};$$

$$C(x \mid y) = \min\{|p| \mid U(p, y) = x\},$$

where U is a fixed universal Turing machine.

Randomness cannot be effectively generated from **scratch**

Randomness cannot be effectively generated from **scratch**

Randomness cannot be effectively generated from **one** string

For any uniformly computable function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, there exists x with $C(x) \geq n - m$ and $C(f(x)) < \log n + O(1)$. So, no uniformly computable function can guarantee an increase of randomness (unless m is very small).

Proof: Let a be the most popular image of f .

Then $C(a) < \log n + O(1)$.

Note that $|f^{-1}(a)| \geq 2^{n-m}$.

So there exists $x \in f^{-1}(a)$ with $C(x) > n - m$.

Randomness cannot be effectively generated from **scratch**

Randomness cannot be effectively generated from **one** string

For any uniformly computable function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, there exists x with $C(x) \geq n - m$ and $C(f(x)) < \log n + O(1)$. So, no uniformly computable function can guarantee an increase of randomness (unless m is very small).

Proof: Let a be the most popular image of f .

Then $C(a) < \log n + O(1)$.

Note that $|f^{-1}(a)| \geq 2^{n-m}$.

So there exists $x \in f^{-1}(a)$ with $C(x) > n - m$.

So we need at least **two** strings to produce better/new randomness

We need to consider the degree of independence of the two strings.

Randomness cannot be effectively generated from **scratch**

Randomness cannot be effectively generated from **one** string

For any uniformly computable function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, there exists x with $C(x) \geq n - m$ and $C(f(x)) < \log n + O(1)$. So, no uniformly computable function can guarantee an increase of randomness (unless m is very small).

Proof: Let a be the most popular image of f .

Then $C(a) < \log n + O(1)$.

Note that $|f^{-1}(a)| \geq 2^{n-m}$.

So there exists $x \in f^{-1}(a)$ with $C(x) > n - m$.

So we need at least **two** strings to produce better/new randomness

We need to consider the degree of independence of the two strings.

DEFINITION: Strings u and v in $\{0, 1\}^n$ have dependency at most $d(n)$, if $C(uv) \geq C(u) + C(v) - d(n)$.

Parameters in this paper

- The input: two finite binary strings x_1 and x_2 of length n .
- x_1 and x_2 have dependency at most d .
- x_1 and x_2 have complexity s .
- The transformation f is a computable function or, better, poly-time computable.
- The output: polynomially many strings $x_3, \dots, x_{n^{k+2}}$ of length m .
- Requirement: Each output string has complexity $\approx m$, even **conditioned by any other output string, or input string.**

Parameters in this paper

- The input: two finite binary strings x_1 and x_2 of length \boxed{n} .
- x_1 and x_2 have dependency at most \boxed{d} .
- x_1 and x_2 have complexity \boxed{s} .
- The transformation f is a computable function or, better, poly-time computable.
- The output: polynomially many strings x_3, \dots, x_{n^k+2} of length \boxed{m} .
- Requirement: Each output string has complexity $\approx m$, even **conditioned by any other output string, or input string.**

Related problem: multi-source extractors.

DEFINITION.

- X_1, \dots, X_k **independent** distributions on $\{0, 1\}^n$,
- each X_i has min-entropy s ,
- $E : (\{0, 1\}^n)^k \rightarrow \{0, 1\}^m$ is a (k -sources, ϵ -biased, s -min-entropy) extractor if

$$|E(X_1, \dots, X_k) - U_m| < \epsilon.$$

Results for multi-source extractors:

- **[CG'88]**: 2 sources, any s , $m \approx s/3$, f computable
- **[BIW'04]**: $\text{poly}(1/\sigma)$ sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[BKSSW'05], [R'06]**: 3-sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[Bou'05]**: 2-sources, $s(n) = (1/2 - (\text{small const.}))n$, $m = \Theta(n)$, f poly-time

Results for multi-source extractors:

- **[CG'88]**: 2 sources, any s , $m \approx s/3$, f computable
- **[BIW'04]**: $\text{poly}(1/\sigma)$ sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[BKSSW'05], [R'06]**: 3-sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[Bou'05]**: 2-sources, $s(n) = (1/2 - (\text{small const.}))n$, $m = \Theta(n)$, f poly-time

Compare with this paper: We want 2 sources, they can have some dependency, poly-many output strings, each one random even conditioned by one source, or any other source, f computable.

Results for multi-source extractors:

- **[CG'88]**: 2 sources, any s , $m \approx s/3$, f computable
- **[BIW'04]**: $\text{poly}(1/\sigma)$ sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[BKSSW'05], [R'06]**: 3-sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[Bou'05]**: 2-sources, $s(n) = (1/2 - (\text{small const.}))n$, $m = \Theta(n)$, f poly-time

Compare with this paper: We want 2 sources, they can have some dependency, poly-many output strings, each one random even conditioned by one source, or any other source, f computable.

- **[Zim'09]**: 2 sources x_1, x_2 , $s(n) = \Omega(\log n)$, with dependency d , $m \approx s/2 - d$, one output string z with $C(z \mid x_i) \approx m - d$

Results for multi-source extractors:

- **[CG'88]**: 2 sources, any s , $m \approx s/3$, f computable
- **[BIW'04]**: $\text{poly}(1/\sigma)$ sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[BKSSW'05], [R'06]**: 3-sources, $s(n) = \sigma n$ -any constant σ , $m = \Theta(n)$, f poly-time
- **[Bou'05]**: 2-sources, $s(n) = (1/2 - (\text{small const.}))n$, $m = \Theta(n)$, f poly-time

Compare with this paper: We want 2 sources, they can have some dependency, **poly-many output strings**, each one random even conditioned by one source, or any other source, f computable.

- **[Zim'09]**: 2 sources x_1, x_2 , $s(n) = \Omega(\log n)$, with dependency d , $m \approx s/2 - d$, **one output string** z with $C(z | x_i) \approx m - d$

The 2 sources are random

Theorem 1

For every k , there is a poly-time function f that on input two n -bit strings x_1, x_2 , produces n -bit strings x_3, \dots, x_{n^k+2} such that for all d if

- $C(x_1) \geq n - \log n$, $C(x_2) \geq n - \log n$,
- x_1 and x_2 are at most d -dependent

then

- $C(x_i) \geq n - d - (k + O(1)) \log n$, for $i \in \{3, \dots, n^k + 2\}$,
- $x_1, x_2, x_3, \dots, x_{n^k+2}$ are pairwise at most $d + (3k + O(1)) \log n$ dependent.

Proof: Generalization of the observation that $z = x_1 \oplus x_2$ is random conditioned by each of the x_j .

The 2 sources have complexity $s < n$

Theorem 2

For every k and every $s(n) = \Omega(k \log n)$, there is a computable function f that on input two n -bit strings x_1, x_2 , produces m -bit strings x_3, \dots, x_{n^k+2} with length $m \approx s/3$ such that if

- $C(x_1) \geq s, C(x_2) \geq s,$
- x_1 and x_2 are at most d -dependent

then

- $C(x_i) \geq m - d - O(\log n),$ for $i \in \{3, \dots, n^k + 2\},$
- $x_1, x_2, x_3, \dots, x_{n^k+2}$ are pairwise at most $d + (2k + O(1)) \log n$ dependent.

Proof: later.

The 2 sources have complexity $O(n)$

Theorem 3

For every $\delta > 0$ and every d , there is a poly-time computable function f that on input two n -bit strings x_1, x_2 , produces one m -bit string x_3 with length $m = \Omega(\delta n)$ such that if

- $C(x_1) \geq \delta n, C(x_2) \geq \delta n,$
- x_1 and x_2 are at most d -dependent

then

- $C(x_3) \geq m - d - \text{poly log } n.$

Proof: Relies on a recent 2-source condenser of Anup Rao [**Rao'08**].

Proof of Th 2. There is a computable function that on input two strings x_1 and x_2 of length n , with complexity s and dependency d , outputs strings x_3, \dots, x_{n^k} of length $m \approx s(n)/3$ such that $K(x_i | x_j) \succeq m - d - O(k \log n)$.

Balanced tables

$T : \{0, 1\}^n \times \{0, 1\}^n \rightarrow$
 $\{0, 1\}^m$, viewed as a coloring
of an

$N \times N$ - table with M colors

($N = 2^n, M = 2^m$).

	u_1	u_2							u_N
u_1									
u_2									
\cdot									
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
u_N									

Balanced tables

$T : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^m$, viewed as a coloring of an

$N \times N$ - table with M colors
($N = 2^n$, $M = 2^m$).

The table is (S, n^k) -balanced if for every colors $a, b \subseteq [M]$, and for every $B_1, B_2 \subseteq [N]$ with $|B_1| \geq S, |B_2| \geq S$, the number of a -colored cells in the $B_1 \times B_2$ rectangle is $\leq \frac{2}{M} |B_1 \times B_2|$ and for every $(i, j) \in [n^k]^2$ the number of cells in the rectangle whose shifted i and j cells are (a, b) colored is $\leq \frac{2}{M^2} |B_1 \times B_2|$.

	u_1	u_2	\cdot	\cdot	\cdot	\cdot	u_N	
u_1								
u_2								
\cdot								
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
u_N								

Balanced tables

$T : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^m$, viewed as a coloring of an

$N \times N$ - table with M colors
($N = 2^n$, $M = 2^m$).

The table is (S, n^k) -balanced if for every colors $a, b \subseteq [M]$, and for every $B_1, B_2 \subseteq [N]$ with $|B_1| \geq S, |B_2| \geq S$, the number of a -colored cells in the $B_1 \times B_2$ rectangle is $\leq \frac{2}{M} |B_1 \times B_2|$ and for every $(i, j) \in [n^k]^2$ the number of cells in the rectangle whose shifted i and j cells are (a, b) colored is $\leq \frac{2}{M^2} |B_1 \times B_2|$.

	u_1	u_2	\cdot	\cdot	\cdot	\cdot	u_N	
u_1								
u_2								
\cdot								
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\cdot
u_N								

NOTE: If all color appear the same number of times in the rectangle, then each color has $\frac{1}{M} |B_1 \times B_2|$ occurrences.

If $M = o((1/\sqrt{n})S^{1/2})$, an (S, n^k) -balanced table can be obtained with the probabilistic method + brute-force search.

proof Th 2 (cont.)

- Build $T : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^m$, $m \approx s/3$, $S = 2^{2s(n)/3}$, (S, n^k) -balanced.
- Produce the new strings:

$$x_3 = E(x_1 + 3, x_2),$$

$$x_4 = E(x_1 + 4, x_2),$$

$$\dots$$

$$x_{n^k+2} = E(x_1 + n^k + 2, x_2)$$

- Proof Part I: $C(x_i | x_1) \succeq m - d$, $C(x_i | x_2) \succeq m - d$, x_i any new string.
- Proof Part II: $C(x_i | x_j) \succeq m - d$, x_i, x_j distinct new strings.

PROOF PART I: $C(x_3 | x_2) \succeq m - d$.

- $t_1 = C(x_1)$, $t_2 = C(x_2)$, $t_1 \geq s$, $t_2 \geq s$.
- $B_1 = \{u \in \{0, 1\}^n \mid C(u) \leq t_1\}$, $S \leq |B_1| < 2^{t_1+1}$.
- $B_2 = \{u \in \{0, 1\}^n \mid C(u) \leq t_2\}$, $S \leq |B_2| < 2^{t_2+1}$.
- x_3 is balanced in the rectangle $B_1 \times B_2$.
- The number of columns in which x_3 is not balanced is $\leq S$. The index of each such column can be described by x_3 (m bits) and $\log S = 2s/3$ bits.
- Since $C(x_2) > m + 2s/3$, x_3 appears balanced in $B_1 \times \{x_2\}$.
- No occurrences of x_3 in $B_1 \times \{x_2\} \leq (2/2^m)|B_1| = 2^{t_1-m+O(1)}$.
- Given x_2 , x_1 can be described by $C(x_3 | x_2)$ bits $+ t_1 - m + O(\log n)$ bits.
- $t_1 - d \leq C(x_2 | x_1) \leq C(x_3 | x_2) + t_1 - m + O(\log n)$.
- So $C(x_3 | x_2) \geq m - d - O(\log n)$.

	u_1	u_2	·	·	x_2	·	·	u_N
u_1								
u_2								
·								
x	·	·	△	△	△	△	△	·
·	·	·	△	△	△	△	△	·
·	·	·	△	△	△	△	△	·
·	·	·	△	△	△	△	△	·
·	·	·	△	△	△	△	△	·
x_N								

PROOF Part II: $C(x_3 | x_4) \succeq m - d$ (Sketch)

- The number of occurrences of (x_3, x_4) in cells $(3, 4)$ shifted from a cell in $B_1 \times B_2$ is $\leq 2/2^{2m} |B_1 \times B_2|$.
- $2/2^{2m} |B_1 \times B_2| \leq 2^{t_1+t_2-m+O(1)}$.
- Cell (x_1, x_2) has its $(3, 4)$ shift colored (x_3, x_4) .
- So, x_1x_2 can be described with $C(x_3x_4) + t_1 + t_2 - 2m + O(\log n)$ bits.
- $t_1 + t_2 - d \leq C(x_1x_2) \leq C(x_3x_4) + t_1 + t_2 - 2m + O(\log n)$.
- $C(x_3x_4) \geq 2m - d + O(\log n)$
- Conclusion follows after some simple calculations.

	u_1	u_2	x_2	\cdot	\cdot	\cdot	\cdot	u_N
u_1								
u_2								
\cdot								
\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\triangle	\cdot
x_1	\cdot	\cdot	\triangle	\triangle	\triangle	\spadesuit	\spadesuit	\cdot
\cdot	\cdot	\cdot	\triangle	\triangle	\triangle	\triangle	\triangle	\cdot
\cdot								
u_N								

Thank you.