Symmetry of Information: a closer look

Marius Zimand

Towson University

WTCS 2012, Auckland

Motivation: Direct product results for randomness

Another Motivation: later ...

General question:

- x random
- y random
- x and y independent

- **Is xy random?**

Classical information and algorithmical information

- Shannon entropy, $H(X) = \sum_x P(X = x) \log P(X = x)$.
  Length of the minimal average description of random variable X.

- Algorithmical plain complexity, $C(x)$.
  Length of the minimal algorithmical description of individual string x.

- Algorithmical prefix-free complexity, $K(x)$.
  Length of the minimal algorithmical prefix-free description of individual string x.

Algorithmical (a.k.a Kolmogorov) complexity

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

- $\overbrace{0101\ldots01}^{10^{100}}$ has a short description.
- flipping a coin 10^{100} times: $0110001010101100101010001010010 11\ldots100$: description $\approx 10^{100}$ bits.

Algorithmical (a.k.a Kolmogorov) complexity

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

- $\overbrace{0101\ldots01}^{10^{100}}$ has a short description.
- flipping a coin 10^{100} times: $0110001010101100101010001010010 11\ldots100$: description $\approx 10^{100}$ bits.

Definition:

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

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

where U is a fixed universal Turing machine.

Algorithmical (a.k.a Kolmogorov) complexity

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

- $\overbrace{0101\ldots01}^{10^{100}}$ has a short description.
- flipping a coin 10^{100} times: $0110001010101100101010001010010 11\ldots 100$: description $\approx 10^{100}$ bits.

Definition:

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

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

where U is a fixed universal Turing machine.

Definition:

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

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

where U is a fixed **prefix-free** universal Turing machine.

Mutual information

- $I(y : x)$ = quantity of information in y about x.
- Classical information theory: $I(Y : X) = H(X) - H(X \mid Y)$.
- Algorithmical information theory: $I(y : x) = C(x) - C(x \mid y)$.

Symmetry of Information

- Symmetry of Information (classical): $I(x:y) = I(y:x)$.
- Symmetry of Information (algorithmical): $I(x:y) = I(y:x) \pm O(\log n)$.

Symmetry of Information

- Symmetry of Information (classical): $I(x : y) = I(y : x)$.
- Symmetry of Information (algorithmical): $I(x : y) = I(y : x) \pm O(\log n)$.
- For some strings x and y, the $\pm O(\log n)$ is necessary.

Symmetry of Information

- Symmetry of Information (classical): $I(x:y) = I(y:x)$.
- Symmetry of Information (algorithmical): $I(x:y) = I(y:x) \pm O(\log n)$.
- For some strings x and y, the $\pm O(\log n)$ is necessary.
- But for some strings, it is not.
- THEOREM [Z, 2011]. For all strings x and y,

$$I(x:y) \leq I(y:x) + O(\log I(y:x)) + O(C^{(2)}(x \mid n) + C^{(2)}(y \mid n)).$$

Symmetry of Information

- Symmetry of Information (classical): $I(x:y) = I(y:x)$.
- Symmetry of Information (algorithmical): $I(x:y) = I(y:x) \pm O(\log n)$.
- For some strings x and y, the $\pm O(\log n)$ is necessary.
- But for some strings, it is not.
- THEOREM [Z, 2011]. For all strings x and y,

$$I(x:y) \le I(y:x) + O(\log I(y:x)) + O(C^{(2)}(x \mid n) + C^{(2)}(y \mid n)).$$

- COROLLARY. If x and y are random, then

$$I(y:x) = O(1) \text{ IFF } I(x:y) = O(1).$$

Randomness direct product

The key part of the proof is to analyze $C(xy \mid n)$ vs. $C(x \mid n) + C(y \mid x)$.

$w = |C(xy \mid n) - C(x \mid n) - C(y \mid x)|$

We show $w = O(\log I(x : y)) + O(C^{(2)}(x \mid n) + C^{(2)}(y \mid n))$.

We get the **direct product result**:

x random, y random, x and y independent $\Rightarrow xy$ random.

Randomness direct product

The key part of the proof is to analyze $C(xy \mid n)$ vs. $C(x \mid n) + C(y \mid x)$.

$w = |C(xy \mid n) - C(x \mid n) - C(y \mid x)|$

We show $w = O(\log I(x : y)) + O(C^{(2)}(x \mid n) + C^{(2)}(y \mid n))$.

We get the **direct product result**:

x random, y random, x and y independent $\Rightarrow$ xy random.

$C(x \mid n) = n - O(1)$

$I(x : y) = O(1)$

Randomness direct product

The key part of the proof is to analyze $C(xy \mid n)$ vs. $C(x \mid n) + C(y \mid x)$.

$w = |C(xy \mid n) - C(x \mid n) - C(y \mid x)|$

We show $w = O(\log I(x : y)) + O(C^{(2)}(x \mid n) + C^{(2)}(y \mid n))$.

We get the **direct product result**:

x random, y random, x and y independent $\Rightarrow xy$ random.

- $t_x = C(x \mid n), t_y = C(y \mid x), t = C(xy \mid n)$
- We want to estimate $w = t_x + t_y - t$
- The construction uses information $\Lambda = (t_x, t_y, w)$
- Λ can be encoded in a self-delimited way using λ bits, for

$$\lambda \leq 2 \log w + O(\log I(x : y) + C^{(2)}(x \mid n) + C^{(2)}(y \mid n).$$

Build a $2^n \times 2^n$ table, with rows and columns indexed by n-bit strings

Color cell (u, v) with 1 if $C(uv \mid n) \leq t$; 0 otherwise.

S = set of 1-cells

S_u = set of 1-cells in row u

We have $|S| \leq 2^{t+1}$.

Let $2^{m-1} < |S_x| \leq 2^m$.
F = the set of rows with $> 2^{m-1}$ 1's.

We have $|F| < \frac{|S|}{2^{m-1}} \leq 2^{t-m+2}$.

	v_1	v_2			
u_1	1	1	0	...	...
u_2	0	0	1	...	...
.	...				
.	...				
.	...				
x		1	1		1
.	...				
.	...				
.	...				

x is in F; F can be enumerated given information Λ

So: $C(x \mid n, \Lambda) \leq t - m + 2 + O(1)$.

y is in S_x; S_x can be enumerated given x and Λ

So: $C(y \mid x, \Lambda) \leq m + O(1)$.

x is in F; F can be enumerated given information Λ

So: $C(x \mid n, \Lambda) \leq t - m + 2 + O(1)$.

y is in S_x; S_x can be enumerated given x and Λ

So: $C(y \mid x, \Lambda) \leq m + O(1)$.

$C(x \mid n, \Lambda) + C(y \mid x, \Lambda) \leq t + O(1) = t_x + t_y - w + O(1)$.

x is in F; F can be enumerated given information Λ

So: $C(x \mid n, \Lambda) \leq t - m + 2 + O(1)$.

y is in S_x; S_x can be enumerated given x and Λ

So: $C(y \mid x, \Lambda) \leq m + O(1)$.

$C(x \mid n, \Lambda) + C(y \mid x, \Lambda) \leq t + O(1) = t_x + t_y - w + O(1)$.

$t_x - O(\lambda) + t_y - O(\lambda) < t_x + t_y - w + O(1)$.

x is in F; F can be enumerated given information Λ

So: $C(x \mid n, \Lambda) \le t - m + 2 + O(1)$.

y is in S_x; S_x can be enumerated given x and Λ

So: $C(y \mid x, \Lambda) \le m + O(1)$.

$C(x \mid n, \Lambda) + C(y \mid x, \Lambda) \le t + O(1) = t_x + t_y - w + O(1)$.

$t_x - O(\lambda) + t_y - O(\lambda) < t_x + t_y - w + O(1)$.

$w < O(\lambda)$.

x is in F; F can be enumerated given information Λ

So: $C(x \mid n, \Lambda) \leq t - m + 2 + O(1)$.

y is in S_x; S_x can be enumerated given x and Λ

So: $C(y \mid x, \Lambda) \leq m + O(1)$.

$C(x \mid n, \Lambda) + C(y \mid x, \Lambda) \leq t + O(1) = t_x + t_y - w + O(1)$.

$t_x - O(\lambda) + t_y - O(\lambda) < t_x + t_y - w + O(1)$.

$w < O(\lambda)$.

Recall that $\lambda \leq 2 \log w + O(\log I(x:y) + C^{(2)}(x \mid n) + C^{(2)}(y \mid n))$.

We obtain: $w = O(\log I(x:y) + C^{(2)}(x \mid n) + C^{(2)}(y \mid n))$. QED

Randomness direct product for prefix-free complexity

- Direct product theorem for plain complexity: If $C(x \mid n) \geq n - c$, and $C(y \mid x) \geq n - c$, then $C(xy \mid 2n) \geq 2n - O(c)$.

Randomness direct product for prefix-free complexity

If x is random and y is random conditioned by x then xy is random

- Direct product theorem for plain complexity: If $C(x \mid n) \geq n - c$, and $C(y \mid x) \geq n - c$, then $C(xy \mid 2n) \geq 2n - O(c)$.

Randomness direct product for prefix-free complexity

- Direct product theorem for plain complexity: If $C(x \mid n) \geq n - c$, and $C(y \mid x) \geq n - c$, then $C(xy \mid 2n) \geq 2n - O(c)$.
- Does the direct product theorem hold for prefix-free complexity?

Randomness direct product for prefix-free complexity

- Direct product theorem for plain complexity: If $C(x \mid n) \geq n - c$, and $C(y \mid x) \geq n - c$, then $C(xy \mid 2n) \geq 2n - O(c)$.
- Does the direct product theorem hold for prefix-free complexity?
- DEFINITION: x is weakly K-random if $K(x \mid n) \geq n - c$.
- DEFINITION: x is strongly K-random if $K(x \mid n) \geq n + K(n) - c$.

Randomness direct product for prefix-free complexity

- Direct product theorem for plain complexity: If $C(x \mid n) \geq n - c$, and $C(y \mid x) \geq n - c$, then $C(xy \mid 2n) \geq 2n - O(c)$.
- Does the direct product theorem hold for prefix-free complexity?
- DEFINITION: x is weakly K-random if $K(x \mid n) \geq n - c$.
- DEFINITION: x is strongly K-random if $K(x \mid n) \geq n + K(n) - c$.
- THEOREM (Direct product for weak K-randomness) [Z'2012] If $K(x \mid n) \geq n - c$, and $K(y \mid x) \geq n - c$, then $K(xy \mid 2n) \geq 2n - O(c)$.

Randomness direct product for prefix-free complexity

- Direct product theorem for plain complexity: If $C(x \mid n) \geq n - c$, and $C(y \mid x) \geq n - c$, then $C(xy \mid 2n) \geq 2n - O(c)$.
- Does the direct product theorem hold for prefix-free complexity?
- DEFINITION: x is weakly K-random if $K(x \mid n) \geq n - c$.
- DEFINITION: x is strongly K-random if $K(x \mid n) \geq n + K(n) - c$.
- THEOREM (Direct product for weak K-randomness) [Z'2012] If $K(x \mid n) \geq n - c$, and $K(y \mid x) \geq n - c$, then $K(xy \mid 2n) \geq 2n - O(c)$.
- For strong K-randomness, the question is open.

THEOREM (Direct product for weak K-randomness) [Z'2012] If $K(x \mid n) \geq n - c$, and $K(y \mid x) \geq n - c$, then $K(xy \mid 2n) \geq 2n - O(c)$.

PROOF(sketch):

Notation: $\overline{d}$ is a self-delimiting encoding of d.

$S_u(d) = \{v \mid K(uv \mid n) \leq 2n - 2d\}$.

$F(d) = \{u \mid |S_u(d)| \geq 2^{n-d}\}$.

We construct prefix-free program p_1 using conditional information x (basically enumerating $S_x(d)$).

> p_1 on input $\overline{d}\,bin(i)$:
> Check if $bin(i)$ is written on $n - d$ bits. If not, diverge.
> Else, enumerate strings of length n such that $K(xu \mid n) \leq 2n - 2d$; output the i-th such string.

If there is i such that $p_1(d, i)$ outputs u, then $K(u \mid x) \leq n - d + |\overline{d}| < n - c$ (for d sufficiently large).

So there is no i such that $p_1(d, i)$ outputs y.

There are two possible reasons:

(a) $K(xy \mid n) > 2n - 2d$; in this case we are done.

(b) There are 2^{n-d} other strings enumerated before y.

But in case (b), $|S_x(d)| \geq 2^{n-d}$, so $x \in F(d)$.

Note that $|F(d)| \leq \frac{2^{2n-2d}}{2^{n-d}} = 2^{n-d}$.

This implies (after some work), $K(x \mid n) < n - d + |\overline{d}| < n - c$, contradiction.

So only (a) can happen. QED

Randomness direct product for infinite sequences

- Van Lambalgen Theorem: x random, and y random conditioned by $x \Leftrightarrow x \oplus y$ is random.
- random means Martin-Löf random.
- $\Rightarrow$ holds also for Schnorr random and constructive random.

Resource-bounded complexity

- Space-bounded computation
  $$CS^{s(n)}(x) = \min\{|p| \mid U(p) = x \text{ in space } \leq s(|x|)\}$$

- Time-bounded computation
  $$CT^{t(n)}(x) = \min\{|p| \mid U(p) = x \text{ in time } \leq t(|x|)\}$$

Randomness direct product for resource-bounded complexity

- THEOREM (Direct product for space-bounded randomness)

  If $CS^{s(n)}(x \mid n) \geq n - c$, $CS^{s(n)}(y \mid x) \geq n - c$, then $CS^{\alpha s(n)}(xy \mid n) \geq 2n - O(c)$, for some constant $\alpha > 0$.

Randomness direct product for resource-bounded complexity

- THEOREM (Direct product for space-bounded randomness)

  If $CS^{s(n)}(x \mid n) \geq n - c$, $CS^{s(n)}(y \mid x) \geq n - c$, then $CS^{\alpha s(n)}(xy \mid n) \geq 2n - O(c)$, for some constant $\alpha > 0$.

- Randomness direct product for time-bounded complexity does not hold (provided one-way permutations exist).

  Take y - random, and $x = f(y)$ where f is a one-way permutation. $CT^{poly}(x \mid n) \geq n$, $CT^{poly}(y \mid x) \geq n$, but $CT^{poly}(xy \mid n) \approx n$.

Open problem for time-bounded complexity

Let x and y be such that

$CT^{poly}(x \mid y) \geq n$, $CT^{poly}(y \mid x) \geq n$.

What can we say about $CT^{poly}(xy \mid n)$?

Conjecture: For some x and y, $CT^{poly}(xy \mid n) << 2n$.

The other motivation...

- Random objects are wonderful (think of random graphs, random strings, ...)
- Randomness is very useful in practice (polls, cryptography, algorithms, games, etc.)
- But how do we get randomness?

How do we get randomness ...

- Randomness cannot be obtained from nothing (entropy cannot be increased).

How do we get randomness ...

- Randomness cannot be obtained from nothing (entropy cannot be increased).
- We need to start with some form of randomness.
- Randomness direct product: From 2 n-bit sources of (close to) perfect randomness, we get one $2n$- bit source of (close to) perfect randomness

How do we get randomness ...

- Randomness cannot be obtained from nothing (entropy cannot be increased).
- We need to start with some form of randomness.
- Randomness direct product: From 2 n-bit sources of (close to) perfect randomness, we get one $2n$- bit source of (close to) perfect randomness
- From *imperfect* randomness sources, can we get *better* randomness?

How do we get randomness ...

- Randomness cannot be obtained from nothing (entropy cannot be increased).
- We need to start with some form of randomness.
- Randomness direct product: From 2 n-bit sources of (close to) perfect randomness, we get one $2n$- bit source of (close to) perfect randomness
- From *imperfect* randomness sources, can we get *better* randomness?
- From *imperfect* randomness sources, can we get *better* and *new* randomness?

How do we get randomness ...

- Randomness cannot be obtained from nothing (entropy cannot be increased).
- We need to start with some form of randomness.
- Randomness direct product: From 2 n-bit sources of (close to) perfect randomness, we get one $2n$- bit source of (close to) perfect randomness
- From *imperfect* randomness sources, can we get *better* randomness?
- From *imperfect* randomness sources, can we get *better* and *new* randomness?

How do we get randomness ...

- From *imperfect* randomness sources, to *better* randomness.

How do we get randomness ...

- From *imperfect* randomness sources, to *better* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \ldots, x_t$ sources with partial randomness, $f(x_1, \ldots, x_t)$ has close to full randomness.

How do we get randomness ...

- From *imperfect* randomness sources, to *better* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \ldots, x_t$ sources with partial randomness, $f(x_1, \ldots, x_t)$ has close to full randomness.
- From *imperfect* randomness sources, to *better* and *new* randomness.

How do we get randomness ...

- From *imperfect* randomness sources, to *better* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \dots, x_t$ sources with partial randomness, $f(x_1, \dots, x_t)$ has close to full randomness.

- From *imperfect* randomness sources, to *better* and *new* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \dots, x_t$ sources with partial randomness, $f(x_1, \dots, x_t)$ conditioned by $x_1, \dots, x_{i-1}, x_{i+1}, \dots x_t$ has close to full randomness.

How do we get randomness ...

- From *imperfect* randomness sources, to *better* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \ldots, x_t$ sources with partial randomness, $f(x_1, \ldots, x_t)$ has close to full randomness.
- From *imperfect* randomness sources, to *better* and *new* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \ldots, x_t$ sources with partial randomness, $f(x_1, \ldots, x_t)$ conditioned by $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_t$ has close to full randomness.
- The sources can be modeled by distributions, and randomness quality by min-entropy.

  f is a randomness extractor.

How do we get randomness ...

- From *imperfect* randomness sources, to *better* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \ldots, x_t$ sources with partial randomness, $f(x_1, \ldots, x_t)$ has close to full randomness.

- From *imperfect* randomness sources, to *better* and *new* randomness.
  We want (polynomial-time) computable f such that on input $x_1, \ldots, x_t$ sources with partial randomness, $f(x_1, \ldots, x_t)$ conditioned by $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_t$ has close to full randomness.

- The sources can be modeled by distributions, and randomness quality by min-entropy.

  f is a randomness extractor.

- The sources can be modeled by strings or sequences, and randomness quality by Kolmogorov complexity.

  f is a Kolmogorov extractor.

Kolmogorov extraction from one string

- Kolmogorov extraction from one string is impossible.

THEOREM. If $f : \{0,1\}^n \rightarrow \{0,1\}^m$ is a uniformly computable family of functions, there is a string x with $C(x) > n - m$ and $C(f(x) \mid n) = O(1)$.

Kolmogorov extraction with advice

- Extraction from one source is possible if we have some non-uniform information about the source.
- Question: How much information?
- THEOREM[FHPVW'06] For every rational $\sigma, \epsilon > 0$, there exists f poly-time computable and a constant k such that for every x with rate(x) $\geq \sigma$, $\exists \alpha_x$ of length k such that rate($f(x, \alpha_x)$) $\geq 1 - \epsilon$.

Kolmogorov extraction with advice

- Extraction from one source is possible if we have some non-uniform information about the source.
- Question: How much information?

rate$(x) = C(x)/|x|$

- THEOREM[FHPVW'06] For every rational $\sigma, \epsilon > 0$, there exists f poly-time computable and a constant k such that for every x with rate$(x) \geq \sigma$, $\exists \alpha_x$ of length k such that rate$(f(x, \alpha_x)) \geq 1 - \epsilon$.

Kolmogorov extraction with advice

- Extraction from one source is possible if we have some non-uniform information about the source.
- Question: How much information?
- THEOREM[FHPVW'06] For every rational $\sigma, \epsilon > 0$, there exists f poly-time computable and a constant k such that for every x with rate$(x) \geq \sigma$, $\exists \alpha_x$ of length k such that rate$(f(x, \alpha_x)) \geq 1 - \epsilon$.
- THEOREM[VV'02,Z'11] With constant advice, we cannot obtain rate$(f(x, \alpha_x)) = 1 - o(1)$.

Kolmogorov extraction with advice

- Extraction from one source is possible if we have some non-uniform information about the source.
- Question: How much information?
- THEOREM[FHPVW'06] For every rational $\sigma, \epsilon > 0$, there exists f poly-time computable and a constant k such that for every x with $\text{rate}(x) \geq \sigma$, $\exists \alpha_x$ of length k such that $\text{rate}(f(x, \alpha_x)) \geq 1 - \epsilon$.
- THEOREM[VV'02,Z'11] With constant advice, we cannot obtain $\text{rate}(f(x, \alpha_x)) = 1 - o(1)$.
- THEOREM[Z'11] With $\omega(1)$ advice, we can obtain $\text{rate}(f(x, \alpha_x)) = 1$.

Kolmogorov extraction from two strings

- Describing the sources:

  $\mathsf{dep}(x, y) = \max\{C(x \mid n) - C(x \mid y), C(y \mid n) - C(y \mid x)\}$

  (k, α) sources: $S_{k,\alpha} = \{(x, y) \mid C(x \mid n) \geq k, C(y \mid n) \geq k, \mathrm{dep(x, y)} \leq \alpha\}$

Kolmogorov extraction from two strings

- Describing the sources:
  $\text{dep}(x, y) = \max\{C(x \mid n) - C(x \mid y), C(y \mid n) - C(y \mid x)\}$
  (k, α) sources: $S_{k,\alpha} = \{(x, y) \mid C(x \mid n) \geq k, C(y \mid n) \geq k, \mathrm{dep(x,y)} \leq \alpha\}$

- THEOREM[Z'10] When we extract from sources with $\text{dep}(x, y) = \alpha$, the randomness deficiency of the output must be $\geq \alpha - O(\log \alpha)$.

Kolmogorov extraction from two strings

- THEOREM[Z'10] Let k, α be such that $k \geq \alpha + 7 \log n$.
  There exists a Kolmogorov extractor E such that for all $(x, y) \in S_{k,\alpha}$,
  1. $|E(x, y)| \approx 2k$,
  2. $C(E(x, y) \mid x) \geq 2k - \alpha - O(\log n)$.

Kolmogorov extraction from two strings

- THEOREM[Z'10] Let k, α be such that $k \geq \alpha + 7 \log n$.
  There exists a Kolmogorov extractor E such that for all $(x, y) \in S_{k,\alpha}$,
  1. $|E(x, y)| \approx 2k$,
  2. $C(E(x, y) \mid x) \geq 2k - \alpha - O(\log n)$.

- THEOREM[Z'10] Let k, α be such that $k \geq \alpha + 7 \log n$.
  There exists a Kolmogorov extractor E such that for all $(x, y) \in S_{k,\alpha}$,
  1. $|E(x, y)| \approx k$,
  2. $C(E(x, y) \mid x) \geq k - \alpha - O(\log n)$.
  3. $C(E(x, y) \mid y) \geq k - \alpha - O(\log n)$.

Kolmogorov extraction from two strings

- THEOREM[Z'10] Let k, α be such that $k \geq \alpha + 7 \log n$.
  There exists a Kolmogorov extractor E such that for all $(x, y) \in S_{k,\alpha}$,
  1. $|E(x, y)| \approx 2k$,
  2. $C(E(x, y) \mid x) \geq 2k - \alpha - O(\log n)$.

- THEOREM[Z'10] Let k, α be such that $k \geq \alpha + 7 \log n$.
  There exists a Kolmogorov extractor E such that for all $(x, y) \in S_{k,\alpha}$,
  1. $|E(x, y)| \approx k$,
  2. $C(E(x, y) \mid x) \geq k - \alpha - O(\log n)$.
  3. $C(E(x, y) \mid y) \geq k - \alpha - O(\log n)$.

- THEOREM[Z'10] In the above theorems if $k = \Omega(n)$, then E is poly-time computable (but output length is a constant fraction of k).

Kolmogorov extraction from infinite sequences

- Sources are sequences in $\{0,1\}^\omega$.
- Quality of randomness: effective Hausdorff dimension $\dim(x)$.
  $\dim(x) = \inf \frac{C(x\upharpoonright n)}{n}$

Kolmogorov extraction from infinite sequences

- Sources are sequences in $\{0,1\}^\omega$.

- Quality of randomness: effective Hausdorff dimension $\dim(x)$.
  $\dim(x) = \inf \frac{C(x \upharpoonright n)}{n}$

- Miller's THEOREM [M'2008] Extraction from one source is impossible.
  There exists x with $\dim(x) = 1/2$ such that for every Turing reduction f, dim $(f(x)) \leq 1/2$.

Kolmogorov extraction from infinite sequences

- DEFINITION[CaludeZ.08] $x, y \in \{0,1\}^\omega$ are C-independent if for all n, m $C(x\restriction n\ y\restriction m) \geq C(x\restriction n) + C(y\restriction m) - O(\log n + \log m)$.

- THEOREM[Z'08] Extraction from two C-independent sources is possible.

  For every rational $\sigma > 0$, there exists a tt-reduction f such that for all x, y, if x and y are C-independent, $\dim(x) \geq \sigma$, $\dim(x) \geq \sigma$, then $\dim(f(x,y)) = 1$.

Summary

Randomness direct product:

IF x random, y random, (x, y) independent THEN xy random.

- Holds for strings and plain Kolmogorov complexity randomness $(C(x \mid n) > n - c)$
- Holds for strings and weak prefix-free Kolmogorov complexity randomness $(K(x \mid n) > n - c)$
- Open for strings and strong prefix-free Kolmogorov complexity randomness $(K(x \mid n) > n + K(n) - c)$
- Holds for infinite sequences and Martin-Löf randomness (van Lamabalgen Theorem) (also for Schnorr randomness, constructive randomness)
- Holds for space-bounded Kolmogorov complexity
- Conjecture: Does not hold for poly-time resource bounded Kolmogorov complexity

La Multi Ani, Cris!

Thank you.