Not Every Domain of a Plain Decompressor Contains the Domain of a Prefix-Free One

Mikhail Andreev, Ilya Razenshteyn and Alexander Shen

July 15, 2010

Kolmogorov complexity

Kolmogorov complexity

- A measure of information in a given string.

Kolmogorov complexity

- A measure of information in a given string.
- One can easily define a set of **Martin-Löf random sequences** (the algorithmic law of large numbers, the algorithmic law of iterated logarithm, etc).

Kolmogorov complexity

- A measure of information in a given string.
- One can easily define a set of **Martin-Löf random sequences** (the algorithmic law of large numbers, the algorithmic law of iterated logarithm, etc).
- Analysis of running time of algorithms (Moser-Tardos, constructive LLL).

Kolmogorov complexity

- A measure of information in a given string.
- One can easily define a set of **Martin-Löf random sequences** (the algorithmic law of large numbers, the algorithmic law of iterated logarithm, etc).
- Analysis of running time of algorithms (Moser-Tardos, constructive LLL).
- And so on...

Decompressors

Decompressors

- **A decompressor** is a computable partial function $D\colon \{0,1\}^* \to \{0,1\}^*$.

Decompressors

- **A decompressor** is a computable partial function $D\colon \{0,1\}^* \to \{0,1\}^*$.
- **Kolmogorov complexity** of a string $x \in \{0,1\}^*$ with respect to a decompressor D: $C_D(x) := \min\{|y| \mid D(y) = x\}$.

Decompressors

- **A decompressor** is a computable partial function $D\colon \{0,1\}^* \to \{0,1\}^*$.
- **Kolmogorov complexity** of a string $x \in \{0,1\}^*$ with respect to a decompressor D: $C_D(x) := \min\{|y| \mid D(y) = x\}$.
- There exists **an optimal decompressor** U such that C_U is minimal up to $O(1)$. $C(x) := C_U(x)$ is **a plain complexity** of x.

Decompressors

- **A decompressor** is a computable partial function $D\colon \{0,1\}^* \to \{0,1\}^*$.
- **Kolmogorov complexity** of a string $x \in \{0,1\}^*$ with respect to a decompressor D: $C_D(x) := \min\{|y| \mid D(y) = x\}$.
- There exists **an optimal decompressor** U such that C_U is minimal up to $O(1)$. $C(x) := C_U(x)$ is **a plain complexity** of x.
- A decompressor is called **prefix-free** if its domain is prefix-free.

Decompressors

- **A decompressor** is a computable partial function $D\colon \{0,1\}^* \to \{0,1\}^*$.
- **Kolmogorov complexity** of a string $x \in \{0,1\}^*$ with respect to a decompressor D: $C_D(x) := \min\{|y| \mid D(y) = x\}$.
- There exists **an optimal decompressor** U such that C_U is minimal up to $O(1)$. $C(x) := C_U(x)$ is **a plain complexity** of x.
- A decompressor is called **prefix-free** if its domain is prefix-free.
- There exists **an optimal prefix-free decompressor** V. $K(x) := C_V(x)$ is **a prefix complexity** of x.

Decompressors

- **A decompressor** is a computable partial function $D\colon \{0,1\}^* \to \{0,1\}^*$.
- **Kolmogorov complexity** of a string $x \in \{0,1\}^*$ with respect to a decompressor D: $C_D(x) := \min\{|y| \mid D(y) = x\}$.
- There exists **an optimal decompressor** U such that C_U is minimal up to $O(1)$. $C(x) := C_U(x)$ is **a plain complexity** of x.
- A decompressor is called **prefix-free** if its domain is prefix-free.
- There exists **an optimal prefix-free decompressor** V. $K(x) := C_V(x)$ is **a prefix complexity** of x.
- The difference $K(x) - C(x)$ can be as large as $\log |x|$.

The main question

The main question

- In DLT 2008 paper Calude, Nies, Staiger and Stephan characterized (supersets of) domains of optimal (prefix-free) decompressors.

The main question

- In DLT 2008 paper Calude, Nies, Staiger and Stephan characterized (supersets of) domains of optimal (prefix-free) decompressors.
- One of their results: a recursively enumerable set W is a superset of the domain of an optimal decompressor iff there is a constant c such that $|A \cap \{0,1\}^n| + \ldots + |A \cap \{0,1\}^{n+c}| \geq 2^n$ for every n.

The main question

- In DLT 2008 paper Calude, Nies, Staiger and Stephan characterized (supersets of) domains of optimal (prefix-free) decompressors.
- One of their results: a recursively enumerable set W is a superset of the domain of an optimal decompressor iff there is a constant c such that $|A \cap \{0,1\}^n| + \ldots + |A \cap \{0,1\}^{n+c}| \geq 2^n$ for every n.
- The main question: **does the domain of every optimal decompressor contain the domain of some optimal prefix-free decompressor**?

The main question

- In DLT 2008 paper Calude, Nies, Staiger and Stephan characterized (supersets of) domains of optimal (prefix-free) decompressors.
- One of their results: a recursively enumerable set W is a superset of the domain of an optimal decompressor iff there is a constant c such that $|A \cap \{0,1\}^n| + \ldots + |A \cap \{0,1\}^{n+c}| \geq 2^n$ for every n.
- The main question: **does the domain of every optimal decompressor contain the domain of some optimal prefix-free decompressor**?
- We show that the answer is **'NO'**.

The main question

- In DLT 2008 paper Calude, Nies, Staiger and Stephan characterized (supersets of) domains of optimal (prefix-free) decompressors.
- One of their results: a recursively enumerable set W is a superset of the domain of an optimal decompressor iff there is a constant c such that $|A \cap \{0,1\}^n| + \ldots + |A \cap \{0,1\}^{n+c}| \geq 2^n$ for every n.
- The main question: **does the domain of every optimal decompressor contain the domain of some optimal prefix-free decompressor**?
- We show that the answer is **'NO'**.
- We build an optimal decompressor U such that for every optimal prefix-free decompressor V holds $\mathrm{dom}\, V \not\subseteq \mathrm{dom}\, U$.

The construction

The construction

- We show that there exists a set A such that:
  1. A contains the domain of one specific optimal decompressor,
  2. A does not contain the domain of any optimal **prefix free** decompressor.

The construction

- We show that there exists a set A such that:
  1. A contains the domain of one specific optimal decompressor,
  2. A does not contain the domain of any optimal **prefix free** decompressor.
- The first part is easy. It is sufficient to require from A the following two properties:
  1. A is recursive,
  2. A is sufficiently dense.

The construction

- We show that there exists a set A such that:
  1. A contains the domain of one specific optimal decompressor,
  2. A does not contain the domain of any optimal **prefix free** decompressor.
- The first part is easy. It is sufficient to require from A the following two properties:
  1. A is recursive,
  2. A is sufficiently dense.
- By **sufficiently dense** we mean that for every n $|A \cap \{0,1\}^n| \geq \varepsilon \cdot 2^n$.

The construction

- We show that there exists a set A such that:
  1. A contains the domain of one specific optimal decompressor,
  2. A does not contain the domain of any optimal **prefix free** decompressor.
- The first part is easy. It is sufficient to require from A the following two properties:
  1. A is recursive,
  2. A is sufficiently dense.
- By **sufficiently dense** we mean that for every n $|A \cap \{0,1\}^n| \geq \varepsilon \cdot 2^n$.
- An easy case of a result from [CNSS]. Consider an arbitrary optimal decompressor D. A is recursive, so there exists a computable injective mapping $i\colon \{0,1\}^* \to \{0,1\}^*$ such that:
  1. $i(x) \in A$ for every x,
  2. $|i(x)| \leq |x| + c$, where c is a fixed constant (one can actually take $c := \lceil \log 1/\varepsilon \rceil$).

The construction

- We show that there exists a set A such that:
  1. A contains the domain of one specific optimal decompressor,
  2. A does not contain the domain of any optimal **prefix free** decompressor.
- The first part is easy. It is sufficient to require from A the following two properties:
  1. A is recursive,
  2. A is sufficiently dense.
- By **sufficiently dense** we mean that for every n $|A \cap \{0,1\}^n| \geq \varepsilon \cdot 2^n$.
- An easy case of a result from [CNSS]. Consider an arbitrary optimal decompressor D. A is recursive, so there exists a computable injective mapping $i\colon \{0,1\}^* \to \{0,1\}^*$ such that:
  1. $i(x) \in A$ for every x,
  2. $|i(x)| \leq |x| + c$, where c is a fixed constant (one can actually take $c := \lceil \log 1/\varepsilon \rceil$).
- $D_1(i(x)) := D(x)$.

The construction

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.
- A will be, in some sense, a universal set. For every n consider strings of length n in the set A. They determine some subset in Cantor space, which is open-closed.
  What is needed for A: every open-closed subset of Cantor space of measure at least $1/3$ is represented at some level and even at many subsequent levels

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.
- A will be, in some sense, a universal set. For every n consider strings of length n in the set A. They determine some subset in Cantor space, which is open-closed.
  What is needed for A: every open-closed subset of Cantor space of measure at least $1/3$ is represented at some level and even at many subsequent levels
- Infinite binary tree $\leftrightarrow \{0,1\}^* \leftrightarrow$ cylinders in $\{0,1\}^\omega$

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.
- A will be, in some sense, a universal set. For every n consider strings of length n in the set A. They determine some subset in Cantor space, which is open-closed.
  What is needed for A: every open-closed subset of Cantor space of measure at least $1/3$ is represented at some level and even at many subsequent levels
- Infinite binary tree $\leftrightarrow \{0,1\}^* \leftrightarrow$ cylinders in $\{0,1\}^\omega$
- $\Omega_x = \{\alpha \in \{0,1\}^\omega \mid \alpha \text{ begins with } x\}$

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.
- A will be, in some sense, a universal set. For every n consider strings of length n in the set A. They determine some subset in Cantor space, which is open-closed.
  What is needed for A: every open-closed subset of Cantor space of measure at least $1/3$ is represented at some level and even at many subsequent levels
- Infinite binary tree $\leftrightarrow \{0,1\}^* \leftrightarrow$ cylinders in $\{0,1\}^\omega$
- $\Omega_x = \{\alpha \in \{0,1\}^\omega \mid \alpha \text{ begins with } x\}$
- $P \subseteq \{0,1\}^\omega$ is **basic** if $P = \bigcup_{i=1}^n \Omega_{x_i}$.

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.
- A will be, in some sense, a universal set. For every n consider strings of length n in the set A. They determine some subset in Cantor space, which is open-closed.
  What is needed for A: every open-closed subset of Cantor space of measure at least $1/3$ is represented at some level and even at many subsequent levels
- Infinite binary tree $\leftrightarrow \{0,1\}^* \leftrightarrow$ cylinders in $\{0,1\}^\omega$
- $\Omega_x = \{\alpha \in \{0,1\}^\omega \mid \alpha \text{ begins with } x\}$
- $P \subseteq \{0,1\}^\omega$ is **basic** if $P = \bigcup_{i=1}^{n} \Omega_{x_i}$.
- $A \subseteq \{0,1\}^*$ **represents** P **at level** n if $P = \bigcup_{x \in A \cap \{0,1\}^n} \Omega_x$.

The construction

- It remains to build a recursive set A with the following properties:
  1. A is sufficiently dense,
  2. A does not contain the domain of any optimal prefix-free decompressor.
- A will be, in some sense, a universal set. For every n consider strings of length n in the set A. They determine some subset in Cantor space, which is open-closed.
  What is needed for A: every open-closed subset of Cantor space of measure at least $1/3$ is represented at some level and even at many subsequent levels
- Infinite binary tree $\leftrightarrow \{0,1\}^* \leftrightarrow$ cylinders in $\{0,1\}^\omega$
- $\Omega_x = \{\alpha \in \{0,1\}^\omega \mid \alpha \text{ begins with } x\}$
- $P \subseteq \{0,1\}^\omega$ is **basic** if $P = \bigcup_{i=1}^n \Omega_{x_i}$.
- $A \subseteq \{0,1\}^*$ **represents** P **at level** n if $P = \bigcup_{x \in A \cap \{0,1\}^n} \Omega_x$.
- **Construction of** A**:** for every basic set P of measure at least $1/3$ there are infinitely many n such that A represents P at levels $n, n+1, \ldots, 2n$.

Two games

Two games

- Lemma: if n_i is a computable sequence such that $\sum_i 2^{-n_i} \leq 1$, then $K(i) \leq n_i + O(1)$.

Two games

- Lemma: if n_i is a computable sequence such that $\sum_i 2^{-n_i} \leq 1$, then $K(i) \leq n_i + O(1)$.
- 'Memory allocation' game:
  - Alice: n_i such that $\sum_i 2^{-n_i} \leq 1$ (one by one).
  - Bob: x_i such that $|x_i| = n_i$ and $\{x_i\}$ is prefix-free set (responds on-line).
  - Bob wins if he is able to allocate desired strings.

Theorem: Bob wins.

Two games

Two games

- The modified game:
  - Bob: $\varepsilon > 0$.
  - Alice: makes at most $2/3$ strings of each length forbidden.
  - Alice: n_i such that $\sum_i 2^{-n_i} \leq \varepsilon$ (one by one).
  - Bob: x_i such that $|x_i| = n_i$ and $\{x_i\}$ is prefix-free set. Moreover, x_i must not be forbidden (responds on-line).
  - Bob wins if he is able to allocate desired strings.

Theorem: Alice wins.

Two games

- The modified game:
  - Bob: $\varepsilon > 0$.
  - Alice: makes at most $2/3$ strings of each length forbidden.
  - Alice: n_i such that $\sum_i 2^{-n_i} \leq \varepsilon$ (one by one).
  - Bob: x_i such that $|x_i| = n_i$ and $\{x_i\}$ is prefix-free set. Moreover, x_i must not be forbidden (responds on-line).
  - Bob wins if he is able to allocate desired strings.

  Theorem: Alice wins.
- Technically, we must consider more complicated game, because complexity is defined up to a constant.

Two games

- The modified game:
  - Bob: $\varepsilon > 0$.
  - Alice: makes at most $2/3$ strings of each length forbidden.
  - Alice: n_i such that $\sum_i 2^{-n_i} \leq \varepsilon$ (one by one).
  - Bob: x_i such that $|x_i| = n_i$ and $\{x_i\}$ is prefix-free set. Moreover, x_i must not be forbidden (responds on-line).
  - Bob wins if he is able to allocate desired strings.

  Theorem: Alice wins.
- Technically, we must consider more complicated game, because complexity is defined up to a constant.
- The idea of a proof is the following: Alice wins, and her winning set is more or less A.

Thank you for your attention.