On the optimal compression of sets in P, NP, P/poly, PSPACE/poly

Marius Zimand

Towson University

CCR 2012- Cambridge
The language compression problem

- If $A$ is computably enumerable, then for every $x \in A$

$$C(x) \leq \log |A^{=n}| + O(\log n)$$

- description of $x$: index of $x$ in an enumeration of $A^{=n}$.

- But enumeration is slow.
The language compression problem

- If $A$ is computably enumerable, then for every $x \in A$
  \[ C(x) \leq \log |A^n| + O(\log n) \]

- description of $x$: index of $x$ in an enumeration of $A^n$.

- But enumeration is slow.

- Is there a time-bounded Kolmogorov complexity version of the above fact?
Informal Definition

\( CD^t(x) = \) length of the shortest program that accepts \( x \) and only \( x \) and runs in \( t(|x|) \) time.
Distinguishing complexity [Sipser 83]

Informal Definition

\[ CD^t(x) = \text{length of the shortest program that accepts } x \text{ and only } x \text{ and runs in } t(|x|) \text{ time.} \]

Formal Definition

\[ CD^t(x) = |p|, \text{ } p \text{ is the shortest program such that} \]

\[
\begin{align*}
U(p, x) & = \text{YES,} \\
U(p, y) & = \text{NO, for all } y \neq x \\
U(p, x) & \text{ halts in } t(|p| + |x|) \text{ steps}
\end{align*}
\]

(\(U\) is a universal Turing machine)
Informal Definition

\[ CD^t(x) = \text{length of the shortest program that accepts } x \text{ and only } x \text{ and runs in } t(|x|) \text{ time.} \]

Formal Definition

\[ CD^t(x) = |p|, \text{ } p \text{ is the shortest program such that} \]

\[
\begin{align*}
U(p, x) &= \text{YES,} \\
U(p, y) &= \text{NO, for all } y \neq x \\
U(p, x) &= \text{halts in } t(|p| + |x|) \text{ steps}
\end{align*}
\]

(\( U \) is a universal Turing machine)

\[ CD^{t,A}(x) - U \text{ uses oracle } A. \]

\[ CND^{t,A}(x) - U \text{ is nondeterministic, } CAMD^{t,A}(x) - U \text{ is Arthur-Merlin machine (randomized + nondeterministic), } CBPD^{t,A} - U \text{ is randomized with bounded error.} \]
What is known:

[Buhrman, Fortnow, Laplante, 2001]: For any set $A$, for every $x \in A$

$$CD^{poly,A}(x) \leq 2 \log |A^n| + O(\log n)$$

[Buhrman, Laplante, Miltersen, 2000]: For some sets $A$, 2 is necessary.
What is known:

[Buhrman, Fortnow, Laplante, 2001]: For any set $A$, for every $x \in A$

$$\text{CD}^{\text{poly}, \mathcal{A}}(x) \leq 2 \log |A^{=n}| + O(\log n)$$

[Buhrman, Laplante, Miltersen, 2000]: For some sets $A$, 2 is necessary.
What is known (cont.):

If we allow nonuniformity

[Sipser, 1983] \( \forall A, \exists \text{ advice } w \text{ of length } \text{poly}(n), \ \forall x \in A \)

\[
\text{CD}^{\text{poly},A}(x | w) \leq \log |A^n| + O(\log n)
\]
What is known (cont.):

If we allow nonuniformity

[Sipser, 1983] \( \forall A, \exists \text{ advice } w \text{ of length } \text{poly}(n), \ \forall x \in A \)

\[
\text{CD}^\text{poly},A(x | w) \leq \log |A^{=n}| + O(\log n)
\]

If we allow some error:

[Buhrman, Fortnow, Laplante, 2001]

\( \forall A, \forall \epsilon, \forall x \in A^{=n} \text{ except } \epsilon \text{ fraction,} \)

\[
\text{CD}^\text{poly},A(x) \leq \log |A^{=n}| + O(\log n)
\]
What is known (cont.):

If we allow nondeterminism:

[Buhrman, Lee, van Melkebeek, 2005]

\[ \forall A, \forall x \in A \]

\[ C^\text{poly}_{A}(x) \leq \log |A^{=n}| + O\left(\sqrt{\log |A^{=n}|} + \log n\right) \log n \]
What is known (cont.):

If we allow nondeterminism:

[Buhrman, Lee, van Melkebeek, 2005]
\[
\forall A, \forall x \in A \quad \text{CND}^{\text{poly}, A}(x) \leq \log |A|^n + O((\sqrt{\log |A|^n} + \log n) \log n)
\]

If we allow randomization + nondeterminism:

[Buhrman, Lee, van Melkebeek, 2005]
\[
\forall A, \forall x \in A \quad \text{CAMD}^{\text{poly}, A}(x) \leq \log |A|^n + O(\log^3 n)
\]

If we allow only randomization, compression can fail

[Buhrman, Lee, van Melkebeek, 2005]
\[
\forall n, t, k < c_1 n^{-c_2 \log t}, t, \exists A \text{ with } \log |A| = n, \forall x \in A \quad \text{CBPD}^{t, A}(x) \geq 2 \log |A|^n - c_3 \log n
\]
What is known (cont.):

If we allow nondeterminism:

[Buhrman, Lee, van Melkebeek, 2005]
\[ \forall A, \forall x \in A \]
\[ C^{\text{nondet}, A}(x) \leq \log |A^n| + O((\sqrt{\log |A^n|} + \log n) \log n) \]

If we allow randomization + nondeterminism:

[Buhrman, Lee, van Melkebeek, 2005]
\[ \forall A, \forall x \in A \]
\[ C^{\text{rand+nondet}, A}(x) \leq \log |A^n| + O(\log^3 n) \]

If we allow only randomization, compression can fail

[Buhrman, Lee, van Melkebeek, 2005]
\[ \forall n, t, k < c_1 n - c_2 \log t, t, \exists A \text{ with } \log |A^n| = k, \forall x \in A \]
\[ C^{\text{rand}, A}(x) \geq 2 \log |A^n| - c_3 \]
QUESTION: For what sets $A$, can we get optimal compression:

$$\forall x \in A^n, \ CD_{poly,A}^{}(x) \leq \log|A^n| + O(\log n). \quad (*)$$
QUESTION: For what sets $A$, can we get optimal compression:

$$\forall x \in A^n, \text{CD}^{\text{poly}, A}(x) \leq \log |A^n| + O(\log n). \quad (*)$$

**ANSWER:** Using a reasonable assumption, (*) holds for every $A$ in PSPACE/poly.
Last year (FCT’2011), I used a method using 2 steps.

Step 1: non-explicit extractors made partially explicit using Nisan pseudo-random generator for constant-depth circuits.
Step 2: Nisan-Wigderson pseudo-random generator assuming a ceratin hardness assumption.

Vinodchandran suggested the following simpler proof for Step 1: extractors are replaced by 2-wise independent distributions.
PROOF for \( A \in P/poly \)

\( P/poly = \) class of sets decidable in polynomial time with polynomial advice.
\( = \) class of sets decidable by polynomial-size circuits.
PROOF for $A \in \text{P/poly}$

$\text{P/poly} = \text{class of sets decidable in polynomial time with polynomial advice.}$
$\quad = \text{class of sets decidable by polynomial-size circuits.}$

Let $A \in \text{P/poly}$ and $x \in A^{=n}$.

Let $k = \lceil \log |A^{=n}| \rceil$.
PROOF for $A \in \text{P/poly}$

$\text{P/poly} = \text{class of sets decidable in polynomial time with polynomial advice.}$

$= \text{class of sets decidable by polynomial-size circuits.}$

Let $A \in \text{P/poly}$ and $x \in A^{=n}$.

Let $k = \lceil \log |A^{=n}| \rceil$.

Suppose we find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$, poly-time computable given $|h|$ bits of information, which isolates $x$ in $A$:

$$\forall y \in A^{=n} \setminus \{x\}, \ h(y) \neq h(x).$$

Then, $h$ and $h(x)$ distinguishes $x$ among the strings in $A^{=n}$. 
PROOF for $A \in \text{P/poly}$

$\text{P/poly} = \text{class of sets decidable in polynomial time with polynomial advice.}$

$= \text{class of sets decidable by polynomial-size circuits.}$

Let $A \in \text{P/poly}$ and $x \in A^n$.

Let $k = \lceil \log |A^n| \rceil$.

Suppose we find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$, poly-time computable given $|h|$ bits of information, which isolates $x$ in $A$:

$$\forall y \in A^n \setminus \{x\}, h(y) \neq h(x).$$

Then, $h$ and $h(x)$ distinguishes $x$ among the strings in $A^n$.

$$\text{CD}^{\text{poly}, A}(x) \leq (k + 1) + |h| + O(\log n) = \log |A^n| + |h| + O(\log n).$$
PROOF for $A \in \text{P/poly}$

$\text{P/poly} = \text{class of sets decidable in polynomial time with polynomial advice.} = \text{class of sets decidable by polynomial-size circuits.}$

Let $A \in \text{P/poly}$ and $x \in A^n$.

Let $k = \lceil \log |A^n| \rceil$.

Suppose we find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$, poly-time computable given $|h|$ bits of information, which isolates $x$ in $A$:

$$\forall y \in A^n \setminus \{x\}, h(y) \neq h(x).$$

Then, $h$ and $h(x)$ distinguishes $x$ among the strings in $A^n$.

$\text{CD}^{\text{poly},A}(x) \leq (k + 1) + |h| + O(\log n) = \log |A^n| + |h| + O(\log n)$.

To finish the proof, I need $h$ that isolates $x$ in $A$ and $|h| = O(\log n)$. 
**PROOF for** $A \in P/poly$ (cont.)

**Problem**

$k = \lceil \log |A^{-n}| \rceil$, $x \in A^{-n}$.

Find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$ that isolates $x$ and $|h|$ is $O(\log n)$.
PROOF for $A \in \text{P/poly}$ (cont.)

**Problem**

$k = \lceil \log |A^n| \rceil$, $x \in A^n$.

Find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$ that isolates $x$ and $|h|$ is $O(\log n)$.

If we choose $h$ randomly,

$$\Pr[h(x) = h(y)] = \frac{1}{2^{k+1}} \quad \text{(for any fixed } y \neq x)$$

$$\Pr[\exists y \in A^n \setminus \{x\}, h(x) = h(y)] \leq 2^k \cdot \frac{1}{2^{k+1}} = \frac{1}{2}$$

So, with probability $\geq 1/2$, $h$ isolates $x$.

But $|h| = 2^n \cdot (k + 1)$. 
PROOF for $A \in \text{P/poly}$ (cont.)

Problem

$k = \lceil \log |A^n| \rceil$, $x \in A^n$.
Find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$ that isolates $x$ and $|h|$ is $O(\log n)$.

STEP 1 (reduction using 2-wise distributions):

- $h$ only needs to be 2-wise independent.
Problem

\[ k = \lceil \log |A^{-n}| \rceil, \; x \in A^{-n}. \]

Find \( h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1} \) that isolates \( x \) and \( |h| \) is \( O(\log n) \).

STEP 1 (reduction using 2-wise distributions):

- \( h \) only needs to be 2-wise independent.
- Take \( h \) a random linear function (i.e., a random \( k \)-by-\( n \) matrix).
- \( h \) is 2-wise independent.
PROOF for $A \in \text{P/poly}$ (cont.)

Problem

$k = \lceil \log |A^{-n}| \rceil, \ x \in A^{-n}$.
Find $h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1}$ that isolates $x$ and $|h|$ is $O(\log n)$.

STEP 1 (reduction using 2-wise distributions):

- $h$ only needs to be 2-wise independent.
- Take $h$ a random linear function (i.e., a random $k$-by-$n$ matrix).
- $h$ is 2-wise independent.
- With probability $\geq 1/2$, $h$ isolates $x$.
- $|h| = n \cdot k$.
PROOF for \( A \in \text{P/poly} \) (cont.)

Problem

\[ k = \lceil \log |A^n| \rceil, \ x \in A^n. \]

Find \( h : \{0, 1\}^n \rightarrow \{0, 1\}^{k+1} \) that isolates \( x \) and \( |h| \) is \( O(\log n) \).

STEP 1 (reduction using 2-wise distributions):

- \( h \) only needs to be 2-wise independent.
- Take \( h \) a random linear function (i.e., a random \( k \)-by-\( n \) matrix).
- \( h \) is 2-wise independent.
- With probability \( \geq 1/2 \), \( h \) isolates \( x \).
- \( |h| = n \cdot k \).
- We have reduced \( |h| \) from \( 2^n \cdot (k + 1) \) to \( n \cdot k \).
PROOF for $A \in \text{P/poly}$ (cont.)

Problem

$k = \lceil \log |A^n| \rceil$, $x \in A^n$.
Find $h : \{0, 1\}^n \to \{0, 1\}^k$ that isolates $x$ and $|h|$ is $O(\log n)$.

STEP 2 (reduction using pseudo-random generators - p.r.g.):
Problem

\[ k = \lceil \log |A^n| \rceil, \; x \in A^n. \]
Find \( h : \{0, 1\}^n \rightarrow \{0, 1\}^k \) that isolates \( x \) and \( |h| \) is \( O(\log n) \).

STEP 2 (reduction using pseudo-random generators - p.r.g.):

- A p.r.g. that fools a class of sets \( C \);

\[ g : \{0, 1\}^{c \log m} \rightarrow \{0, 1\}^m, \text{ computable in poly. time in } m \]

such that for every \( B \in C \)

\[ \Pr_{s \in \{0, 1\}^{c \log m}}[g(s) \in B] \approx \epsilon \Pr_{u \in \{0, 1\}^m}[u \in B]. \]

- No set in \( C \) can distinguish between an output of \( g \) and a uniformly generated string.
PROOF for \( A \in \text{P/poly} \) (cont.)

\[ B = \{ h \mid h \text{ linear and } h \text{ does not isolate } x \} \]
PROOF for $A \in \text{P/poly}$ (cont.)

- $B = \{ h \mid h \text{ linear and } h \text{ does not isolate } x \}$
- $B$ is in $\text{NP/poly}$. 

Suppose we have a p.r.g. $g : \{0,1\}^c \log n \rightarrow \{0,1\}^{kn}$ that fools $\text{NP/poly}$ sets. $g$ fools $B$. $B$ is large, so for many $s$, $g(s) \in B$. For some seed $s$ (actually for many seeds), $g(s)$ is an $h$ that isolates $x$. Thus we can compute $h$ from $s$ which has $O(\log n)$ bits. This is exactly what we need.
PROOF for $A \in P/poly$ (cont.)

- $B = \{h \mid h$ linear and $h$ does not isolate $x\}$
- $B$ is in $NP/poly$.
- Suppose we have a p.r.g. $g : \{0, 1\}^{c \log n} \rightarrow \{0, 1\}^{kn}$ that fools $NP/poly$ sets.
- $g$ fools $B$. 
PROOF for $A \in P/poly$ (cont.)

- $B = \{ h \mid h$ linear and $h$ does not isolate $x \}$
- $B$ is in $NP/poly$.
- Suppose we have a p.r.g. $g : \{0,1\}^{c \log n} \rightarrow \{0,1\}^{kn}$ that fools $NP/poly$ sets.
- $g$ fools $B$.
- $\overline{B}$ is large, so for many $s$, $g(s) \in \overline{B}$. 
PROOF for $A \in \text{P/poly}$ (cont.)

- $B = \{h \mid h$ linear and $h$ does not isolate $x\}$
- $B$ is in \text{NP/poly}.
- Suppose we have a p.r.g. $g : \{0, 1\}^{c \log n} \rightarrow \{0, 1\}^{kn}$ that fools \text{NP/poly} sets.
- $g$ fools $B$.
- $\overline{B}$ is large, so for many $s$, $g(s) \in \overline{B}$.
- For some seed $s$ (actually for many seeds), $g(s)$ is an $h$ that isolates $x$.
- Thus we can compute $h$ from $s$ which has $O(\log n)$ bits.
\( B = \{ h \mid h \text{ linear and } h \text{ does not isolate } x \} \)

- \( B \) is in \( \text{NP/poly} \).
- Suppose we have a p.r.g. \( g : \{0, 1\}^{c \log n} \rightarrow \{0, 1\}^{kn} \) that fools \( \text{NP/poly} \) sets.
- \( g \) fools \( B \).
- \( B \) is large, so for many \( s \), \( g(s) \in \overline{B} \).
- For some seed \( s \) (actually for many seeds), \( g(s) \) is an \( h \) that isolates \( x \).
- Thus we can compute \( h \) from \( s \) which has \( O(\log n) \) bits.
- This is exactly what we need.
Pseudo random generators

- How do we get a p.r.g.?
Pseudo random generators

- How do we get a p.r.g.?
- Start with a function $f$ computable in $E = \bigcup_c \text{DTIME}[2^{cn}]$ that is hard.
- How hard? Depends on what sets do we want the p.r.g. to fool.

Assumption H: There exists a function $f$ computable in $E$ that for some $\epsilon > 0$ cannot be computed by circuits with SAT gates of size $2^{\epsilon n}$.

$H \implies$ p.r.g. that fools NP/poly $\implies$ sets in P/poly can be compressed optimally.
Pseudo random generators

- How do we get a p.r.g.?
- Start with a function $f$ computable in $E = \cup_c \text{DTIME}[2^{cn}]$ that is hard.
- How hard? Depends on what sets do we want the p.r.g. to fool.
- To fool sets in NP/poly we need an $f$ that requires circuits with SAT gates of size $2^{\epsilon n}$, for some $\epsilon > 0$. 

Assumption H: There exists a function $f$ computable in $E$ that for some $\epsilon > 0$ cannot be computed by circuits with SAT gates of size $2^{\epsilon n}$.

$H \Rightarrow$ p.r.g. that fools NP/poly $\Rightarrow$ sets in P/poly can be compressed optimally.
Pseudo random generators

- How do we get a p.r.g.?
- Start with a function $f$ computable in $E = \bigcup_c \text{DTIME}[2^{cn}]$ that is **hard**.
- How hard? Depends on what sets do we want the p.r.g. to fool.
- To fool sets in NP/poly we need an $f$ that requires circuits with SAT gates of size $2^{\epsilon n}$, for some $\epsilon > 0$.
- The output of $f$ is somewhat unpredictable, but the p.r.g. requirements are much more demanding.
- Using lots of clever ideas (Nisan, Wigderson, Impagliazzo, Sudan, Trevisan, Vadhan, Klivans, van Melkebeek) from $f$ one can construct a p.r.g $g$ that fools NP/poly.
How do we get a p.r.g.?
Start with a function $f$ computable in $E = \bigcup_c \text{DTIME}[2^{cn}]$ that is hard.
How hard? Depends on what sets do we want the p.r.g. to fool.
To fool sets in NP/poly we need an $f$ that requires circuits with SAT gates of size $2^{\epsilon n}$, for some $\epsilon > 0$.
The output of $f$ is somewhat unpredictable, but the p.r.g. requirements are much more demanding.
Using lots of clever ideas (Nisan, Wigderson, Impagliazzo, Sudan, Trevisan, Vadhan, Klivans, van Melkebeek) from $f$ one can construct a p.r.g $g$ that fools NP/poly.
Assumption H: There exists a function $f$ computable in $E$ that for some $\epsilon > 0$ cannot be computed by circuits with SAT gates of size $2^{\epsilon n}$.
$H \Rightarrow$ p.r.g. that fools NP/poly $\Rightarrow$ sets in P/poly can be compressed optimally.
Our result

Assumption H: There exists a function $f$ computable in $E$ that for some $\epsilon > 0$ cannot be computed by circuits with SAT gates of size $2^{\epsilon n}$.

Theorem

Assume H. For any set $A$ in $P/poly$, there exists a polynomial $p$ such that for every $x \in A$

$$CD_{p,A}(x) \leq \log |A^{=n}| + O(\log n)$$
Similar results for sets in P, NP, $\Sigma^p_k$, PSPACE/poly.
Similar results for sets in $P$, $NP$, $\Sigma^p_k$, $PSPACE/poly$.

For $PSPACE/poly$

Theorem

Assume there exists a function $f$ computable in $E$ but not in $DSPACE[2^{o(n)}]$. For any set $A$ in $PSPACE/poly$, there exists a polynomial $p$ such that for every $x \in A$

$$CD^{p,A}(x) \leq \log |A^{=n}| + O(\log n)$$
• Pseudo-random generators based on similar assumptions have been used before in resource-bounded Kolmogorov complexity.

• (Antunes, Fortnow, 2009) If hardness assumption holds, then $m^p(x) = 2^{-C^p(x)}$ is universal among P-samplable distributions.

For any P-samplable distribution $\sigma$, there is a polynomial $p$ such that $C^p(x) \leq \log 1/\sigma(x) + O(\log n)$.

• (Antunes, Fortnow, Pinto, Souza, 2007) Computational depth cannot grow fast.
How to show $P \neq NP$

Find a set $A$ such that

1. $\mathsf{CD} \text{poly}, A(x) \geq 2\log|A| = n$, for some $x \in A$ (like [Buhrman, Laplante, Miltersen])

2. $\mathsf{CD} \text{poly}, \Sigma^P_k \oplus A(x) \leq (2 - \epsilon) \log|A| = n$, for all $x \in A$

Then, $\Sigma^P_k \neq P$.

It is reasonable to try $A$ in the Polynomial Hierarchy. But $\mathsf{PH} \subseteq \mathsf{PSPACE}$, so (1) will not succeed. So look for $A$ outside $\mathsf{PSPACE}$. 

Marius Zimand (Towson U.)
How to show $P \neq NP$

Find a set $A$ such that

(1) $\text{CD}^{\text{poly},A}(x) \geq 2 \log |A^n|$, for some $x \in A$ (like [Buhrman, Laplante, Miltersen])

(2) $\text{CD}^{\text{poly},\Sigma^p_k \oplus A}(x) \leq (2 - \epsilon) \log |A^n|$, for all $x \in A$

Then, $\Sigma^p_k \neq P$. 
How to show $P \neq NP$

Find a set $A$ such that

(1) $\text{CD}^{\text{poly},A}(x) \geq 2 \log |A|=n|$, for some $x \in A$ (like [Buhrman,Laplante, Miltersen] )

(2) $\text{CD}^{\text{poly},\Sigma^p_k \oplus A}(x) \leq (2 - \epsilon) \log |A|=n|$, for all $x \in A$

Then, $\Sigma^p_k \neq P$.

It is reasonable to try $A$ in the Polynomial Hierarchy.

But $\text{PH} \subseteq \text{PSPACE}$, so (1) will not succeed.

So look for $A$ outside PSPACE.
Thank you.