The Isomorphism Conjecture

Manindra Agrawal

IIT Kanpur

CiE, Heidelberg, July 22, 2009

Outline

The Isomorphism Theorem for c.e. Class

Theorem (Myhill, 1955)

Let A and B be two $\leq_m$-complete sets for c.e.. Then $A \equiv B$.

$\leq_m$-complete	:	many-one complete
$\equiv$	:	isomorphic under computable isomorphisms

Translation to the Class NP

Isomorphism Conjecture (Berman-Hartmanis, 1977)

Let A and B be two $\leq_m^p$-comoplete sets for NP. Then $A \equiv^p B$.

$\leq_m^p$-complete	:	many-one polynomial-time complete
$\equiv^p$	:	isomorphic under polynomial-time computable and invertible isomorphisms ($\equiv$ p-isomorphic)

Proof of the Isomorphism Theorem for c.e. Class

Proof is in two steps:

Theorem (Step 1)

A $\leq_m$-complete set for c.e. is also $\leq_1$-complete.

Theorem (Step 2)

Let A and B be two $\leq_1$-complete sets for c.e.. Then $A \equiv B$.

$\leq_1$-complete : one-one complete

Proof of the Isomorphism Theorem for c.e. Class

Proof is in two steps:

Theorem (Step 1)

A $\leq_m$-complete set for c.e. is also $\leq_1$-complete.

Theorem (Step 2)

Let A and B be two $\leq_1$-complete sets for c.e.. Then $A \equiv B$.

$\leq_1$-complete : one-one complete

Translating Steps to NP

- For a 1-1 function f, $|f(x)|$ can sometimes be much smaller than $|x|$.
- So an isomorphism between two $\leq_1^p$-complete sets for NP may require large time complexity.
- To avoind this, we need to have $\leq_{1,si}^p$-completeness.
- Then the isomorphism can be computed in NP.

¡2-¿ $\leq_{1,si}^p$-complete : complete under 1-1, size-increasing polynomial-time reductions

Translating Steps to NP

- For a 1-1 function f, $|f(x)|$ can sometimes be much smaller than $|x|$.
- So an isomorphism between two $\leq_1^p$-complete sets for NP may require large time complexity.
- To avoid this, we need to have $\leq_{1,si}^p$-completeness.
- Then the isomorphism can be computed in NP.

¡2-¿ $\leq_{1,si}^p$-complete : complete under 1-1, size-increasing polynomial-time reductions

An Alternative Conjecture

Weak Isomorphism Conjecture

Let A and B be two $\leq_m^p$-comoplete sets for NP. Then $A \equiv_w^p B$.

$\equiv_w^p$	:	isomorphic under polynomial-time computable and NP-invertible isomorphisms
NP-computable	:	computed by polynomial-time single-valued NTMs

An Alternative Conjecture

- Sufficient to prove that all $\leq_m^p$-complete sets for NP are also $\leq_{1,si}^p$-complete.
- Hence, would be easier to prove.
- Lacks symmetry: one direction of isomorphism is easier to compute than the other.
- For this reason, Isomorphism Conjecture is considered the "right" translation.

An Alternative Conjecture

- Sufficient to prove that all $\leq_m^p$-complete sets for NP are also $\leq_{1,si}^p$-complete.
- Hence, would be easier to prove.
- Lacks symmetry: one direction of isomorphism is easier to compute than the other.
- For this reason, Isomorphism Conjecture is considered the "right" translation.

An Alternative Conjecture

- Sufficient to prove that all $\leq_m^p$-complete sets for NP are also $\leq_{1,si}^p$-complete.
- Hence, would be easier to prove.
- Lacks symmetry: one direction of isomorphism is easier to compute than the other.
- For this reason, Isomorphism Conjecture is considered the “right” translation.

Outline

Partial Results

Theorem (Berman-Hartmanis, 1977)

Let A and B be two $\leq^p_{1,si,i}$-complete sets for NP. Then $A \equiv^p B$.

Mimics step 2 of Myhill's proof.

$\leq^p_{1,si,i}$-complete : complete under 1-1, size-increasing, and polynomial-time invertible reductions

Partial Results

Theorem (Berman-Hartmanis, 1977)

Let A and B be two $\leq^p_{1,si,i}$-complete sets for NP. Then $A \equiv^p B$.

Mimics step 2 of Myhill's proof.

$\leq^p_{1,si,i}$-complete : complete under 1-1, size-increasing, and polynomial-time invertible reductions

Partial Results

To Prove

A $\leq_m^p$-complete set for NP is also $\leq_{1,si,i}^p$-complete.

(Berman-Hartmanis, 1977) showed that all known $\leq_m^p$-complete sets for NP at the time were $\leq_{1,si,i}^p$-complete.

Partial Results

To Prove

A $\leq_m^p$-complete set for NP is also $\leq_{1,si,i}^p$-complete.

(Berman-Hartmanis, 1977) showed that all known $\leq_m^p$-complete sets for NP at the time were $\leq_{1,si,i}^p$-complete.

Proving is Hard

Observation

If the Isomorphism Conjecture holds then $P \neq NP$.

Proof Sketch.

If the conjecture holds that all $\leq_m^p$-complete sets are dense and there are sparse sets in P.

Definition

Set A is dense if there exists an $\epsilon > 0$ such that for every n: $|A|_{\leq n} \geq 2^{n^\epsilon}$.
Set A is sparse if there exists a polynomial p such that for every n: $|A|_{\leq n} \leq p(n)$.

Proving is Hard

Observation

If the Isomorphism Conjecture holds then P $\neq$ NP.

Proof Sketch.

If the conjecture holds that all $\leq_m^p$-complete sets are dense and there are sparse sets in P.

Definition

Set A is dense if there exists an $\epsilon > 0$ such that for every n: $|A|_{\leq n} \geq 2^{n^\epsilon}$.
Set A is sparse if there exists a polynomial p such that for every n: $|A|_{\leq n} \leq p(n)$.

Proof Assuming P $\neq$ NP is also Hard

Theorem (Kurtz, 1983)

There is an oracle A such that $P^A \neq NP^A$ and the Isomorphism Conjecture is false relative to A.

Theorem (Fenner-Fortnow-Kurtz, 1994)

There is an oracle B such that the Isomorphism Conjecture is true relative to B.

Proof Assuming P $\neq$ NP is also Hard

Theorem (Kurtz, 1983)

There is an oracle A such that $P^A \neq NP^A$ and the Isomorphism Conjecture is false relative to A.

Theorem (Fenner-Fortnow-Kurtz, 1994)

There is an oracle B such that the Isomorphism Conjecture is true relative to B.

Proving A Consequence

At least the following consequence can be proved assuming $P \neq NP$:

Theorem (Mahaney, 1982)

If $P \neq NP$ then no $\leq_m^p$-complete set for NP is sparse.

Outline

One-Way Functions

Definition

Function g is $s(n)$-secure one-way function if

- g is polynomial-time computable, and
- for every probabilistic polynomial-time TM M and for every n:

$$\Pr_{x \in_R \{0,1\}^n}[f(M(f(x))) = f(x)] < \frac{1}{s(n)}.$$

An $s(n)$-secure one-way function can be inverted efficiently only on $\frac{1}{s(n)}$ fraction of strings of size n.

One-Way Functions

Definition

Function g is $s(n)$-secure one-way function if

- g is polynomial-time computable, and
- for every probabilistic polynomial-time TM M and for every n:

$$\Pr_{x \in_R \{0,1\}^n}[f(M(f(x))) = f(x)] < \frac{1}{s(n)}.$$

An $s(n)$-secure one-way function can be inverted efficiently only on $\frac{1}{s(n)}$ fraction of strings of size n.

One-Way Functions

Definition

Function g is $s(n)$-secure one-way function if

- g is polynomial-time computable, and
- for every probabilistic polynomial-time TM M and for every n:

$$\Pr_{x \in_R \{0,1\}^n} [f(M(f(x))) = f(x)] < \frac{1}{s(n)}.$$

An $s(n)$-secure one-way function can be inverted efficiently only on $\frac{1}{s(n)}$ fraction of strings of size n.

Examples of One-Way Functions

Multiplication: $g_M(x, y) = x \cdot y$.

- Believed to be $1 + \frac{1}{n^5}$-secure.

Exponentiation in a finite field: $g_E(e, g, p) = (g^e \pmod{p}, g, p)$.

- Believed to be $1 + \frac{1}{n^4}$-secure.

Both are believed to be $1 + \frac{1}{n^{O(1)}}$-secure even if the inverting TM is allowed 2^{n^ϵ} time for some small $\epsilon > 0$.

Examples of One-Way Functions

Multiplication: $g_M(x, y) = x \cdot y$.

- Believed to be $1 + \frac{1}{n^5}$-secure.

Exponentiation in a finite field: $g_E(e, g, p) = (g^e \ (mod\ p), g, p)$.

- Believed to be $1 + \frac{1}{n^4}$-secure.

Both are believed to be $1 + \frac{1}{n^{O(1)}}$-secure even if the inverting TM is allowed 2^{n^ϵ} time for some small $\epsilon > 0$.

Examples of One-Way Functions

Multiplication: $g_M(x, y) = x \cdot y$.

- Believed to be $1 + \frac{1}{n^5}$-secure.

Exponentiation in a finite field: $g_E(e, g, p) = (g^e \ (mod\ p), g, p)$.

- Believed to be $1 + \frac{1}{n^4}$-secure.

Both are believed to be $1 + \frac{1}{n^{O(1)}}$-secure even if the inverting TM is allowed 2^{n^ϵ} time for some small $\epsilon > 0$.

Increasing Security

Theorem

Let $g = g_M$ or g_E. Define

$$\hat{g}(x_1 x_2 \cdots x_m) = g(x_1) g(x_2) \cdots g(x_m)$$

where $|x_1| = |x_2| = \cdots = |x_m| = n$ and $m = |x_1|^{2/\epsilon}$. Then $\hat{g}$ is $2^{n^{\epsilon/2}}$-secure.

Thus, any probabilistic polynomial-time TM can invert $\hat{g}$ on at most $\frac{2^n}{2^{n^{\epsilon/2}}}$ strings of size n.

Increasing Security

Theorem

Let $g = g_M$ or g_E. Define

$$\hat{g}(x_1 x_2 \cdots x_m) = g(x_1) g(x_2) \cdots g(x_m)$$

where $|x_1| = |x_2| = \cdots = |x_m| = n$ and $m = |x_1|^{2/\epsilon}$. Then $\hat{g}$ is $2^{n^{\epsilon/2}}$-secure.

Thus, any probabilistic polynomial-time TM can invert $\hat{g}$ on at most $\frac{2^n}{2^{n^{\epsilon/2}}}$ strings of size n.

Encrypted NP-complete Set

- Let g be a 1-1, size-increasing, 2^{n^ϵ}-secure one-way function.
- Consider set
$$g(\mathrm{SAT}) = \{g(x) \mid x \in \mathrm{SAT}\}$$
where SAT is the set of all satisfiable boolean formulas.
- Since g is 1-1, and size-increasing, $g(\mathrm{SAT})$ is $\leq^p_{1,si}$-complete for NP.
- Since g is hard to invert almost everywhere, $g(\mathrm{SAT})$ may not be $\leq^p_{1,si,i}$-complete.
- Studied by (Joseph-Young, 1985).

Encrypted NP-complete Set

- Let g be a 1-1, size-increasing, 2^{n^ϵ}-secure one-way function.
- Consider set

$$g(\mathrm{SAT}) = \{g(x) \mid x \in \mathrm{SAT}\}$$

where SAT is the set of all satisfiable boolean formulas.
- Since g is 1-1, and size-increasing, $g(\mathrm{SAT})$ is $\leq^p_{1,si}$-complete for NP.
- Since g is hard to invert almost everywhere, $g(\mathrm{SAT})$ may not be $\leq^p_{1,si,i}$-complete.
- Studied by (Joseph-Young, 1985).

Encrypted NP-complete Set

- Let g be a 1-1, size-increasing, 2^{n^ϵ}-secure one-way function.
- Consider set
$$g(\mathsf{SAT}) = \{g(x) \mid x \in \mathsf{SAT}\}$$
where SAT is the set of all satisfiable boolean formulas.
- Since g is 1-1, and size-increasing, $g(\mathsf{SAT})$ is $\leq^p_{1,si}$-complete for NP.
- Since g is hard to invert almost everywhere, $g(\mathsf{SAT})$ may not be $\leq^p_{1,si,i}$-complete.
- Studied by (Joseph-Young, 1985).

Encrypted NP-complete Set

- Let g be a 1-1, size-increasing, 2^{n^ϵ}-secure one-way function.
- Consider set

$$g(\mathsf{SAT}) = \{g(x) \mid x \in \mathsf{SAT}\}$$

  where SAT is the set of all satisfiable boolean formulas.
- Since g is 1-1, and size-increasing, $g(\mathsf{SAT})$ is $\leq^p_{1,si}$-complete for NP.
- Since g is hard to invert almost everywhere, $g(\mathsf{SAT})$ may not be $\leq^p_{1,si,i}$-complete.
- Studied by (Joseph-Young, 1985).

Encrypted NP-complete Set

Encrypted Complete Set Conjecture (Joseph-Young, 1985)

There exists a 1-1, size-increasing one-way function g such that $g(\text{SAT})$ is not $\leq^p_{1,si,i}$-complete for NP.

If the conjecture is true then the Isomorphism Conjecture is false.

Encrypted NP-complete Set

Encrypted Complete Set Conjecture (Joseph-Young, 1985)

There exists a 1-1, size-increasing one-way function g such that $g(\mathrm{SAT})$ is not $\leq^p_{1,si,i}$-complete for NP.

If the conjecture is true then the Isomorphism Conjecture is false.

Scrambling Functions

Definition

Function g is a scrambling function if

- g is 1-1, size-increasing and polynomial-time computable, and
- there is no dense polynomial-time subset of range(g).

Observation

Scrambling functions are $2^{n-n^{o(1)}}$-secure one-way functions if we restrict to invertibility by deterministic polynomial-time TMs.

Scrambling Functions

Definition

Function g is a scrambling function if

- g is 1-1, size-increasing and polynomial-time computable, and
- there is no dense polynomial-time subset of range(g).

Observation

Scrambling functions are $2^{n-n^{o(1)}}$-secure one-way functions if we restrict to invertibility by deterministic polynomial-time TMs.

Scrambling Functions

Definition

Function g is a scrambling function if

- g is 1-1, size-increasing and polynomial-time computable, and
- there is no dense polynomial-time subset of range(g).

Observation

Scrambling functions are $2^{n-n^{o(1)}}$-secure one-way functions if we restrict to invertibility by deterministic polynomial-time TMs.

Scrambling Functions

Theorem (Kurtz-Mahaney-Royer, 1989)

If scrambling functions exist then the Encrypted Complete Set Conjecture is true.

Proof Sketch.

If SAT $\leq^p_{1,si,i}$ g(SAT) via h then $h(X)$ is a dense polynomial-time subset of range(g) for X a dense set in SAT $\cap$ P.

Evidence for Scrambling Functions

Theorem (Kurtz-Mahaney-Royer, 1989)

Scrambling functions exist relative to a random oracle.

- Theorefore, the Isomorphism Conjecture is false relative to a random oracle.
- However, it is not clear if scrambling functions exist in the real world.

Evidence for Scrambling Functions

Theorem (Kurtz-Mahaney-Royer, 1989)

Scrambling functions exist relative to a random oracle.

- Theorefore, the Isomorphism Conjecture is false relative to a random oracle.
- However, it is not clear if scrambling functions exist in the real world.

Evidence for Scrambling Functions

Theorem (Kurtz-Mahaney-Royer, 1989)

Scrambling functions exist relative to a random oracle.

- Theorefore, the Isomorphism Conjecture is false relative to a random oracle.
- However, it is not clear if scrambling functions exist in the real world.

Outline

Outline

EXP

Theorem (Berman, 1977)

All $\leq_m^p$-complete sets for EXP are also $\leq_{1,si}^p$-complete.

- Hence, the Weak Isomorphism Conjecture is true for EXP.
- Similar for higher deterministic classes.

EXP

Theorem (Berman, 1977)

All $\leq_m^p$-complete sets for EXP are also $\leq_{1,si}^p$-complete.

- Hence, the Weak Isomorphism Conjecture is true for EXP.
- Similar for higher deterministic classes.

NEXP

Theorem (Ganesan-Homer, 1989)

All $\leq^p_m$-complete sets for NEXP are also $\leq^p_1$-complete.

Similar for higher deterministic classes.

Outline

An Isomorphic Degree

Theorem (Kurtz-Mahaney-Royer 1988)

There is a many-one degree inside 2-tt-complete degree of EXP such that all sets in the degree are p-isomorphic to each other.

2-tt complete degree : class of complete sets under truth-table reductions that make at most 2 queries.

A Non-isomorphic Degree

Theorem (Ko-Long-Du 1987)

If $P \neq UP$ then there is a 1-1, size-increasing degree inside 2-tt-complete degree of EXP containing two sets that are not p-isomorphic to each other.

- UP is the class of sets accepted by polynomial-time NTMs that have at most one accepting path on any input.
- P $\neq$ UP iff there exist 1-1, size-increasing, 1-secure one-way functions.

A Non-isomorphic Degree

Theorem (Ko-Long-Du 1987)

If $P \neq UP$ then there is a 1-1, size-increasing degree inside 2-tt-complete degree of EXP containing two sets that are not p-isomorphic to each other.

- UP is the class of sets accepted by polynomial-time NTMs that have at most one accepting path on any input.
- $P \neq UP$ iff there exist 1-1, size-increasing, 1-secure one-way functions.

A CHARECTERIZATION

COROLLARY (KO-LONG-DU 1987)

$P \neq UP$ iff there is a 1-1, size-increasing degree that is not a p-isomorphic degree.

Outline

1-NL Functions

Definition

Function f is a 1-NL function if there exists a NTM with a one-way input tape and work tape space bounded by $O(\log n)$ that computes f.

Theorem (A 1994)

Let A and B be two $\leq_m^{1-NL}$-complete sets for NP. Then $A \equiv^{1-NL} B$.

1-NL Functions

Definition

Function f is a 1-NL function if there exists a NTM with a one-way input tape and work tape space bounded by $O(\log n)$ that computes f.

Theorem (A 1994)

Let A and B be two $\leq_m^{1-NL}$-complete sets for NP. Then $A \equiv^{1-NL} B$.

AC^0 Functions

Definition

Function f is a AC^0 function if there exists a (uniform) polynomial-size, constant depth circuit family that computes f.

Theorem (AAR 1996, AAIPR 1997, A 2000, A 2001)

Let A and B be two $\leq_m^{AC^0}$-complete sets for NP. Then $A =^{AC^0} B$.

All "natural" $\leq_m^p$-complete sets for NP are also $\leq_m^{AC^0}$-complete.

AC^0 Functions

Definition

Function f is a AC^0 function if there exists a (uniform) polynomial-size, constant depth circuit family that computes f.

Theorem (AAR 1996, AAIPR 1997, A 2000, A 2001)

Let A and B be two $\leq_m^{AC^0}$-complete sets for NP. Then $A \equiv^{AC^0} B$.

All "natural" $\leq_m^p$-complete sets for NP are also $\leq_m^{AC^0}$-complete.

AC^0 Functions

Definition

Function f is a AC^0 function if there exists a (uniform) polynomial-size, constant depth circuit family that computes f.

Theorem (AAR 1996, AAIPR 1997, A 2000, A 2001)

Let A and B be two $\leq_m^{AC^0}$-complete sets for NP. Then $A \equiv^{AC^0} B$.

All "natural" $\leq_m^p$-complete sets for NP are also $\leq_m^{AC^0}$-complete.

Some Observations

- Translated to AC^0 settings, there exist $2^{(\log n)^{O(1)}}$-secure, 1-1, size-increasing one-way functions.
  - Function is computed by uniform AC^0 circuit family and is secure against polynomial-size non-uniform AC^0 circuits of depth d (for some d).
- Yet, on a dense subset, these functions can be inverted by a depth two AC^0 circuit.

Some Observations

- Translated to AC^0 settings, there exist $2^{(\log n)^{O(1)}}$-secure, 1-1, size-increasing one-way functions.
  - Function is computed by uniform AC^0 circuit family and is secure against polynomial-size non-uniform AC^0 circuits of depth d (for some d).
- Yet, on a dense subset, these functions can be inverted by a depth two AC^0 circuit.

Some Observations

- The proof of Isomorphism Theorem for AC^0 uses pseudorandom generators, a stronger form of one-way functions.
- So the one-way functions here help proving the conjecture!

Some Observations

- The proof of Isomorphism Theorem for AC^0 uses pseudorandom generators, a stronger form of one-way functions.
- So the one-way functions here help proving the conjecture!

Outline

Using One-way Functions

Theorem (A 2003, A-Watanabe 2009)

Suppose there exist 2^{n^ϵ}-secure, 1-1, size-increasing one-way functions. Then $\leq_m^p$-complete sets for NP are also $\leq_{1,si}^{P/poly}$-complete.

$\leq_{1,si}^{P/poly}$-complete : computable by polynomial-time TMs that have polynomial-sized advice available.

Proof Idea

- Use one-way function to construct a 1-1, size-increasing pseudorandom generator, say h.
- Let A be any $\leq_m^p$-complete set for NP.
- Let $h(\mathrm{SAT} \times \{0,1\}^*)$ reduce to A via g.
- g must be 1-1 and size-increasing on most inputs otherwise it contradicts pseudorandomness of h.
- Now define a reduction f of SAT to $\mathrm{SAT} \times \{0,1\}^*$ as: $f(x) = (x, R)$ where R is a random string, $|R|$ a large polynomial in $|x|$.
- For most of R, $g \circ h \circ f$ is a 1-1, size-increasing reduction of SAT to A.
- Fixing an appropriate R for each length gives the result.

Proof Idea

- Use one-way function to construct a 1-1, size-increasing pseudorandom generator, say h.
- Let A be any $\leq_m^p$-complete set for NP.
- Let $h(\mathrm{SAT} \times \{0,1\}^*)$ reduce to A via g.
- g must be 1-1 and size-increasing on most inputs otherwise it contradicts pseudorandomness of h.
- Now define a reduction f of SAT to $\mathrm{SAT} \times \{0,1\}^*$ as: $f(x) = (x, R)$ where R is a random string, $|R|$ a large polynomial in $|x|$.
- For most of R, $g \circ h \circ f$ is a 1-1, size-increasing reduction of SAT to A.
- Fixing an appropriate R for each length gives the result.

Proof Idea

- Use one-way function to construct a 1-1, size-increasing pseudorandom generator, say h.
- Let A be any $\leq_m^p$-complete set for NP.
- Let $h(\mathsf{SAT} \times \{0,1\}^*)$ reduce to A via g.
- g must be 1-1 and size-increasing on most inputs otherwise it contradicts pseudorandomness of h.
- Now define a reduction f of SAT to $\mathsf{SAT} \times \{0,1\}^*$ as: $f(x) = (x, R)$ where R is a random string, $|R|$ a large polynomial in $|x|$.
- For most of R, $g \circ h \circ f$ is a 1-1, size-increasing reduction of SAT to A.
- Fixing an appropriate R for each length gives the result.

Proof Idea

- Use one-way function to construct a 1-1, size-increasing pseudorandom generator, say h.
- Let A be any $\leq_m^p$-complete set for NP.
- Let $h(\mathsf{SAT} \times \{0,1\}^*)$ reduce to A via g.
- g must be 1-1 and size-increasing on most inputs otherwise it contradicts pseudorandomness of h.
- Now define a reduction f of SAT to $\mathsf{SAT} \times \{0,1\}^*$ as: $f(x) = (x, R)$ where R is a random string, $|R|$ a large polynomial in $|x|$.
- For most of R, $g \circ h \circ f$ is a 1-1, size-increasing reduction of SAT to A.
- Fixing an appropriate R for each length gives the result.

Proof Idea

- Use one-way function to construct a 1-1, size-increasing pseudorandom generator, say h.
- Let A be any $\leq_m^p$-complete set for NP.
- Let $h(\mathrm{SAT} \times \{0,1\}^*)$ reduce to A via g.
- g must be 1-1 and size-increasing on most inputs otherwise it contradicts pseudorandomness of h.
- Now define a reduction f of SAT to $\mathrm{SAT} \times \{0,1\}^*$ as: $f(x) = (x, R)$ where R is a random string, $|R|$ a large polynomial in $|x|$.
- For most of R, $g \circ h \circ f$ is a 1-1, size-increasing reduction of SAT to A.
- Fixing an appropriate R for each length gives the result.

One-way Functions with Easy Cylinders

Definition

Let f be a 1-1, size-increasing, P/poly-computable function. Function f has an easy cylinder if

- There is a P/poly-computable embedding function e computable by circuits of size $q(n)$ on inputs of size n,
- There exist polynomial $\ell(n)$ with $\ell(n) \geq q(n)$,
- For every n, for every u, $|u| = \ell(n)$, there exists P/poly-computable function g_u such that

$$g_u(f(u, e(x))) = x$$

for all x, $|x| = n$.

One-way Functions with Easy Cylinders

Definition

Let f be a 1-1, size-increasing, P/poly-computable function. Function f has an easy cylinder if

- There is a P/poly-computable embedding function e computable by circuits of size $q(n)$ on inputs of size n,
- There exist polynomial $\ell(n)$ with $\ell(n) \geq q(n)$,
- For every n, for every u, $|u| = \ell(n)$, there exists P/poly-computable function g_u such that

$$g_u(f(u, e(x))) = x$$

for all x, $|x| = n$.

One-way Functions with Easy Cylinders

Definition

Let f be a 1-1, size-increasing, P/poly-computable function. Function f has an easy cylinder if

- There is a P/poly-computable embedding function e computable by circuits of size $q(n)$ on inputs of size n,
- There exist polynomial $\ell(n)$ with $\ell(n) \geq q(n)$,
- For every n, for every u, $|u| = \ell(n)$, there exists P/poly-computable function g_u such that

$$g_u(f(u, e(x))) = x$$

for all x, $|x| = n$.

One-way Functions with Easy Cylinders

- A function f with an an easy cylinder has parameterized (by u) p-invertible subsets.
- Given n and u of length $\ell(n)$, the embedding function e maps $\{0,1\}^n$ to $u \cdot \{0,1\}^{\leq q(n)}$ such that f is invertible on $u \cdot e(\{0,1\}^n)$.
- The embedding function e is independent of u but the inverting function g_u is allowed to depend on u.
- The definition can be generalized to allow e also to be (moderately) dependent on u.

One-way Functions with Easy Cylinders

- A function f with an an easy cylinder has parameterized (by u) p-invertible subsets.
- Given n and u of length $\ell(n)$, the embedding function e maps $\{0,1\}^n$ to $u \cdot \{0,1\}^{\leq q(n)}$ such that f is invertible on $u \cdot e(\{0,1\}^n)$.
- The embedding function e is independent of u but the inverting function g_u is allowed to depend on u.
- The definition can be generalized to allow e also to be (moderately) dependent on u.

One-way Functions with Easy Cylinders

- A function f with an an easy cylinder has parameterized (by u) p-invertible subsets.
- Given n and u of length $\ell(n)$, the embedding function e maps $\{0,1\}^n$ to $u \cdot \{0,1\}^{\leq q(n)}$ such that f is invertible on $u \cdot e(\{0,1\}^n)$.
- The embedding function e is independent of u but the inverting function g_u is allowed to depend on u.
- The definition can be generalized to allow e also to be (moderately) dependent on u.

One-way Functions with Easy Cylinders

- A function f with an an easy cylinder has parameterized (by u) p-invertible subsets.
- Given n and u of length $\ell(n)$, the embedding function e maps $\{0,1\}^n$ to $u \cdot \{0,1\}^{\leq q(n)}$ such that f is invertible on $u \cdot e(\{0,1\}^n)$.
- The embedding function e is independent of u but the inverting function g_u is allowed to depend on u.
- The definition can be generalized to allow e also to be (moderately) dependent on u.

One-way Functions with Easy Cylinders

Theorem (A-Watanabe 2009)

Let f be a 1-1, size-increasing, P/poly-computable function with an easy cylinder. Then $K \equiv^{P/poly} f(K)$.

K	:	a special set with $K \equiv^{p}$ SAT
$\equiv^{P/poly}$	:	isomorphic via P/poly-computable and invertible isomorphisms

Some Functions with Easy Cylinders: Multiplication

$g_M(xy) = x \cdot y$, $|x| = |y|$.

- Let $\ell(n) = n$, $e(y) = y$.
- Fixing a u, $|u| = n = |y|$, g_M becomes $g_M^u(y) = u \cdot y$.
- This is easily invertible.

Some Functions with Easy Cylinders: Multiplication

$g_M(xy) = x \cdot y$, $|x| = |y|$.

- Let $\ell(n) = n$, $e(y) = y$.
- Fixing a u, $|u| = n = |y|$, g_M becomes $g_M^u(y) = u \cdot y$.
- This is easily invertible.

Some Functions with Easy Cylinders: Multiplication

$g_M(xy) = x \cdot y$, $|x| = |y|$.

- Let $\ell(n) = n$, $e(y) = y$.
- Fixing a u, $|u| = n = |y|$, g_M becomes $g_M^u(y) = u \cdot y$.
- This is easily invertible.

Some Functions with Easy Cylinders: Exponentiation

$g_E(egp) = (g^e \pmod p), g, p)$.

- Let $\ell(n) = 2n$, $e(y) = y$.
- Fixing a $u = eg$, $|u| = 2n = 2|y|$, g_E becomes $g_E^u(y) = (g^e \pmod y), g, y)$.
- This is easily invertible.

Some Functions with Easy Cylinders: Exponentiation

$g_E(egp) = (g^e \ (mod \ p), g, p)$.

- Let $\ell(n) = 2n$, $e(y) = y$.
- Fixing a $u = eg$, $|u| = 2n = 2|y|$, g_E becomes $g_E^u(y) = (g^e \ (mod \ y), g, y)$.
- This is easily invertible.

Some Functions with Easy Cylinders: Exponentiation

$g_E(egp) = (g^e \ (mod\ p), g, p)$.

- Let $\ell(n) = 2n$, $e(y) = y$.
- Fixing a $u = eg$, $|u| = 2n = 2|y|$, g_E becomes $g_E^u(y) = (g^e \ (mod\ y), g, y)$.
- This is easily invertible.

Some Functions with Easy Cylinders: Concatenation

Let $g_C(x_1 x_2 \cdots x_m) = g(x_1)g(x_2)\cdots g(x_m)$ for $|x_1| = |x_2| = \cdots = |x_m|$ and g a function with easy cylinder.

- Concatenation increases security.
- Let $\ell(n) = (m-1)n$, $e(y) = y$.
- Fixing a $u = x_1 x_2 \cdots x_{m-1}$, $|u| = (m-1)n = (m-1)|y|$, g_C becomes $g_C^u(y) = g(x_1)g(x_2)\cdots g(x_{m-1})g(y)$.
- Now use the easy cylinder property of g.

Some Functions with Easy Cylinders: Concatenation

Let $g_C(x_1 x_2 \cdots x_m) = g(x_1)g(x_2)\cdots g(x_m)$ for $|x_1| = |x_2| = \cdots = |x_m|$ and g a function with easy cylinder.

- Concatenation increases security.
- Let $\ell(n) = (m-1)n$, $e(y) = y$.
- Fixing a $u = x_1 x_2 \cdots x_{m-1}$, $|u| = (m-1)n = (m-1)|y|$, g_C becomes $g_C^u(y) = g(x_1)g(x_2)\cdots g(x_{m-1})g(y)$.
- Now use the easy cylinder property of g.

Some Functions with Easy Cylinders: Concatenation

Let $g_C(x_1x_2\cdots x_m) = g(x_1)g(x_2)\cdots g(x_m)$ for $|x_1| = |x_2| = \cdots = |x_m|$ and g a function with easy cylinder.

- Concatenation increases security.
- Let $\ell(n) = (m-1)n$, $e(y) = y$.
- Fixing a $u = x_1x_2\cdots x_{m-1}$, $|u| = (m-1)n = (m-1)|y|$, g_C becomes $g_C^u(y) = g(x_1)g(x_2)\cdots g(x_{m-1})g(y)$.
- Now use the easy cylinder property of g.

Some Functions with Easy Cylinders: Concatenation

Let $g_C(x_1 x_2 \cdots x_m) = g(x_1)g(x_2)\cdots g(x_m)$ for $|x_1| = |x_2| = \cdots = |x_m|$ and g a function with easy cylinder.

- Concatenation increases security.
- Let $\ell(n) = (m-1)n$, $e(y) = y$.
- Fixing a $u = x_1 x_2 \cdots x_{m-1}$, $|u| = (m-1)n = (m-1)|y|$, g_C becomes $g_C^u(y) = g(x_1)g(x_2)\cdots g(x_{m-1})g(y)$.
- Now use the easy cylinder property of g.

A Conjecture

Easy Cylinder Conjecture

Every 1-1, size-increasing, P/poly-computable function has an easy cylinder.

Implies that all $\leq_m^{P/poly}$-complete sets for NP are P/poly-isomorphic.

A Conjecture

Easy Cylinder Conjecture

Every 1-1, size-increasing, P/poly-computable function has an easy cylinder.

Implies that all $\leq_m^{P/poly}$-complete sets for NP are P/poly-isomorphic.

Is It True?

- For most of the known one-way functions, it is easy to show they have easy cylinder.
- For P/poly-computable one-way functions, it is more involved.
- If g has an easy cylinder, how about g^n? For example, g_M^n?

Is It True?

- For most of the known one-way functions, it is easy to show they have easy cylinder.
- For P/poly-computable one-way functions, it is more involved.
- If g has an easy cylinder, how about g^n? For example, g_M^n?

Is It True?

- For most of the known one-way functions, it is easy to show they have easy cylinder.
- For P/poly-computable one-way functions, it is more involved.
- If g has an easy cylinder, how about g^n? For example, g_M^n?

Eliminating Nonuniformity?

- Transforming $\leq_m^p$-compleness to $\leq_{1,si}^{P/poly}$-completeness, the nonuniformity is due to choice of R in the function $f(x) = (x, R)$.
- A random choice works with high probability.
- Can one find a deterministic way to choose R?
- Transforming $\leq_{1,si}^{p}$-compleness to $\leq_{1,si,i}^{P/poly}$-completeness, the nonuniformity is due to:
  - choice of e, and
  - choice of g_u.
- If e is uniform and there is a polynomial-time mapping from u to g_u, this step becomes uniform.

Eliminating Nonuniformity?

- Transforming $\leq_m^p$-compleness to $\leq_{1,si}^{P/poly}$-completeness, the nonuniformity is due to choice of R in the function $f(x) = (x, R)$.
- A random choice works with high probability.
- Can one find a deterministic way to choose R?
- Transforming $\leq_{1,si}^p$-compleness to $\leq_{1,si,i}^{P/poly}$-completeness, the nonuniformity is due to:
  - choice of e, and
  - choice of g_u.
- If e is uniform and there is a polynomial-time mapping from u to g_u, this step becomes uniform.

Eliminating Nonuniformity?

- Transforming $\leq_m^p$-compleness to $\leq_{1,si}^{P/poly}$-completeness, the nonuniformity is due to choice of R in the function $f(x) = (x, R)$.
- A random choice works with high probability.
- Can one find a deterministic way to choose R?
- Transforming $\leq_{1,si}^p$-compleness to $\leq_{1,si,i}^{P/poly}$-completeness, the nonuniformity is due to:
  - choice of e, and
  - choice of g_u.
- If e is uniform and there is a polynomial-time mapping from u to g_u, this step becomes uniform.

Thank You!